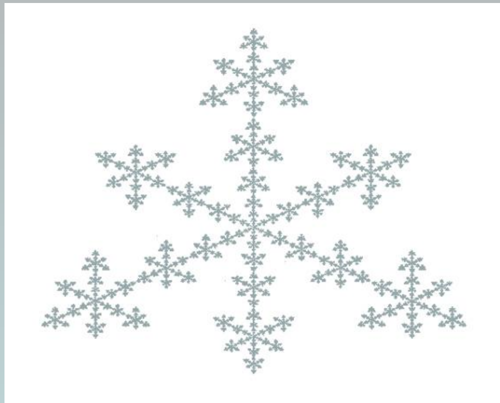
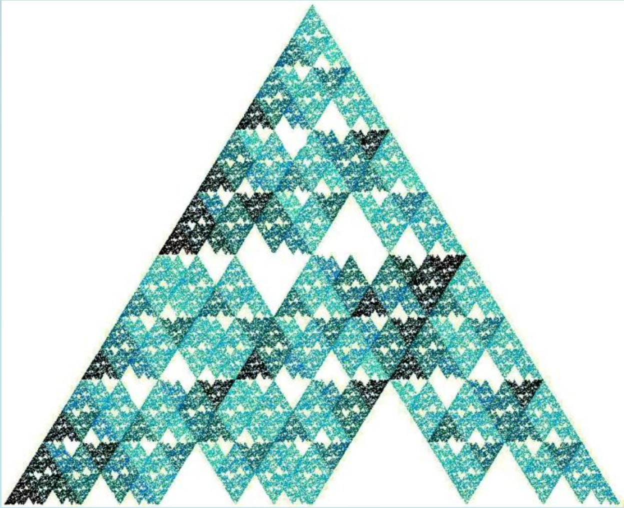
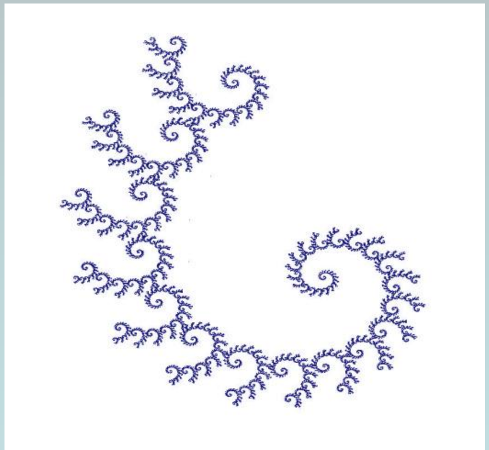
A Taste of π

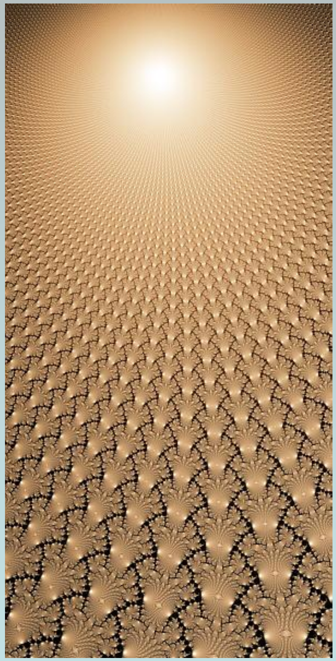
Fractals and the Chaos Game

Randall Pyke
Senior Lecturer
Department of Mathematics, SFU

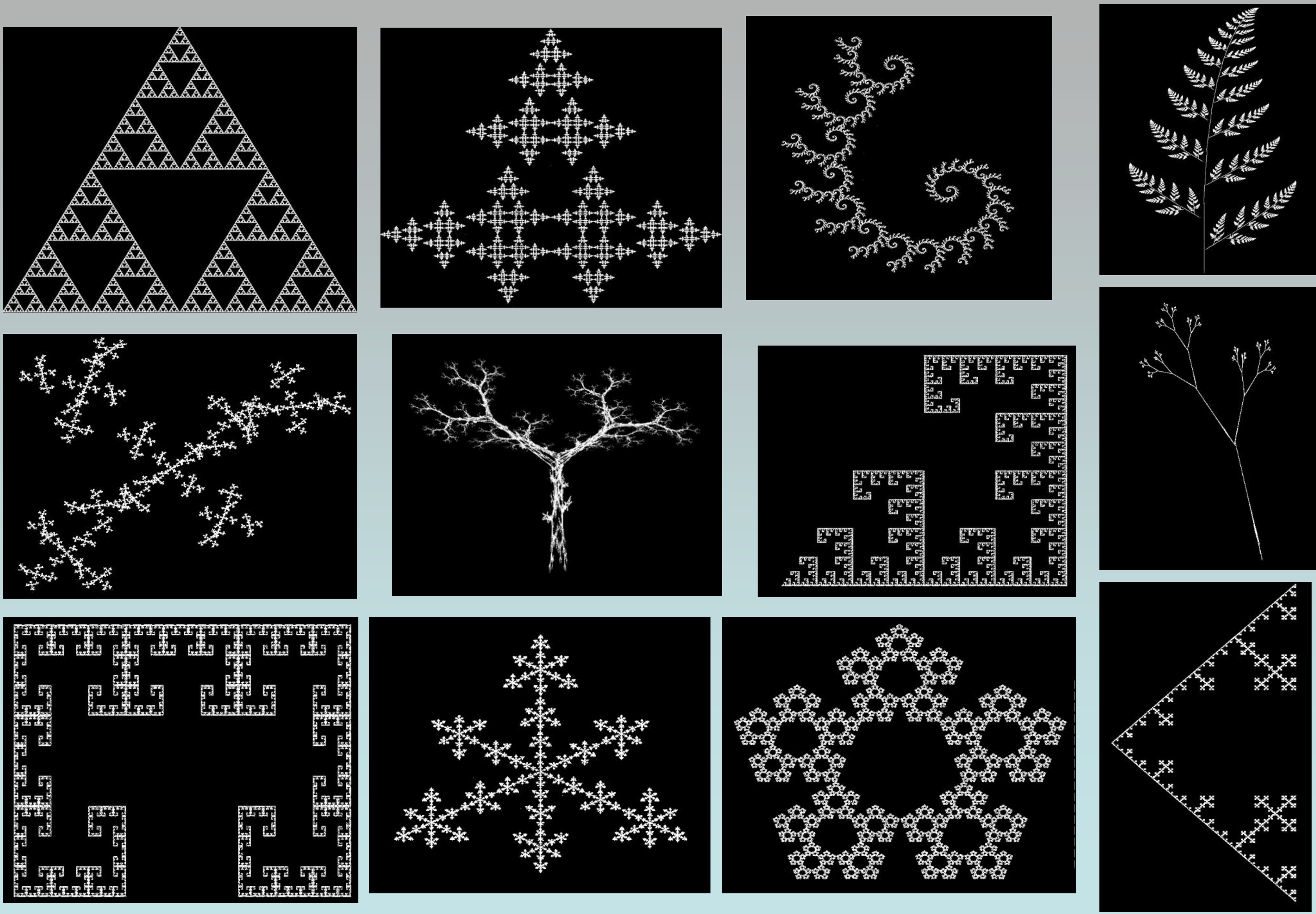








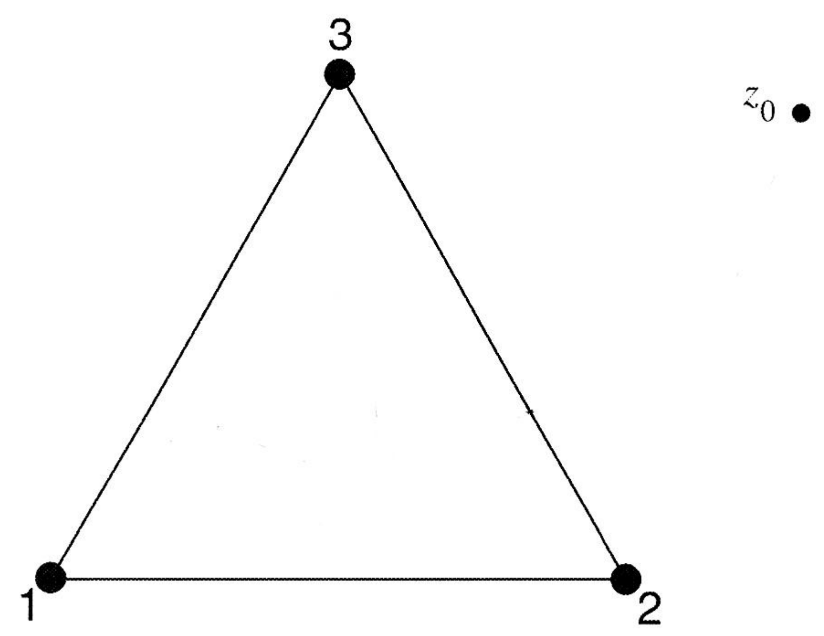
Fractals



A Game…….

Sierpinski (Triangle)

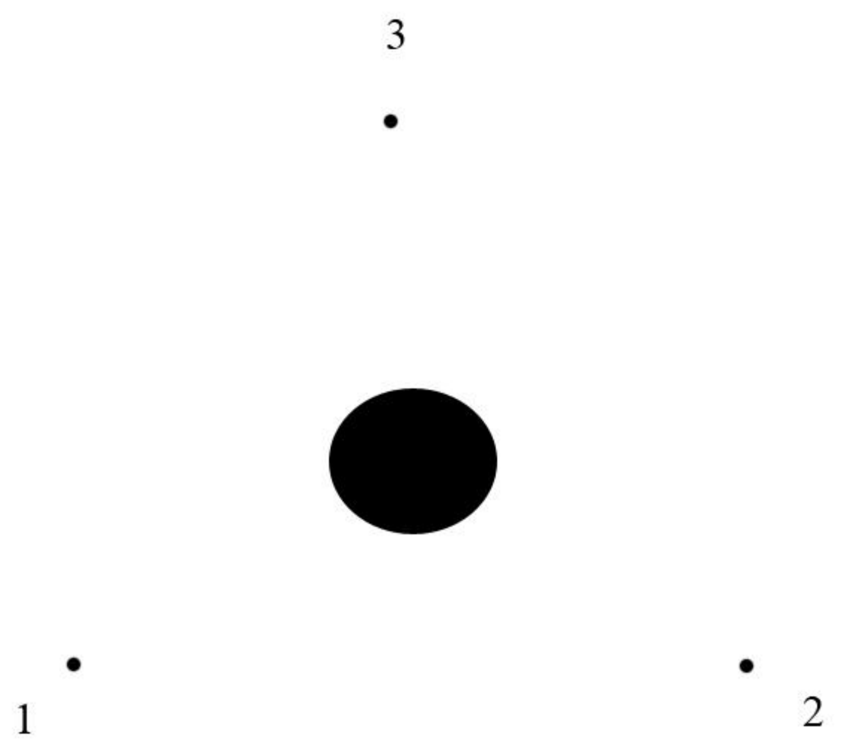
- **three pins 1, 2, 3, arranged at vertices of equilateral triangle**
- **choose random number s_i from** $\{1, 2, 3\}$
- **move 1/2 distance from current game point to black pin labelled s_i**

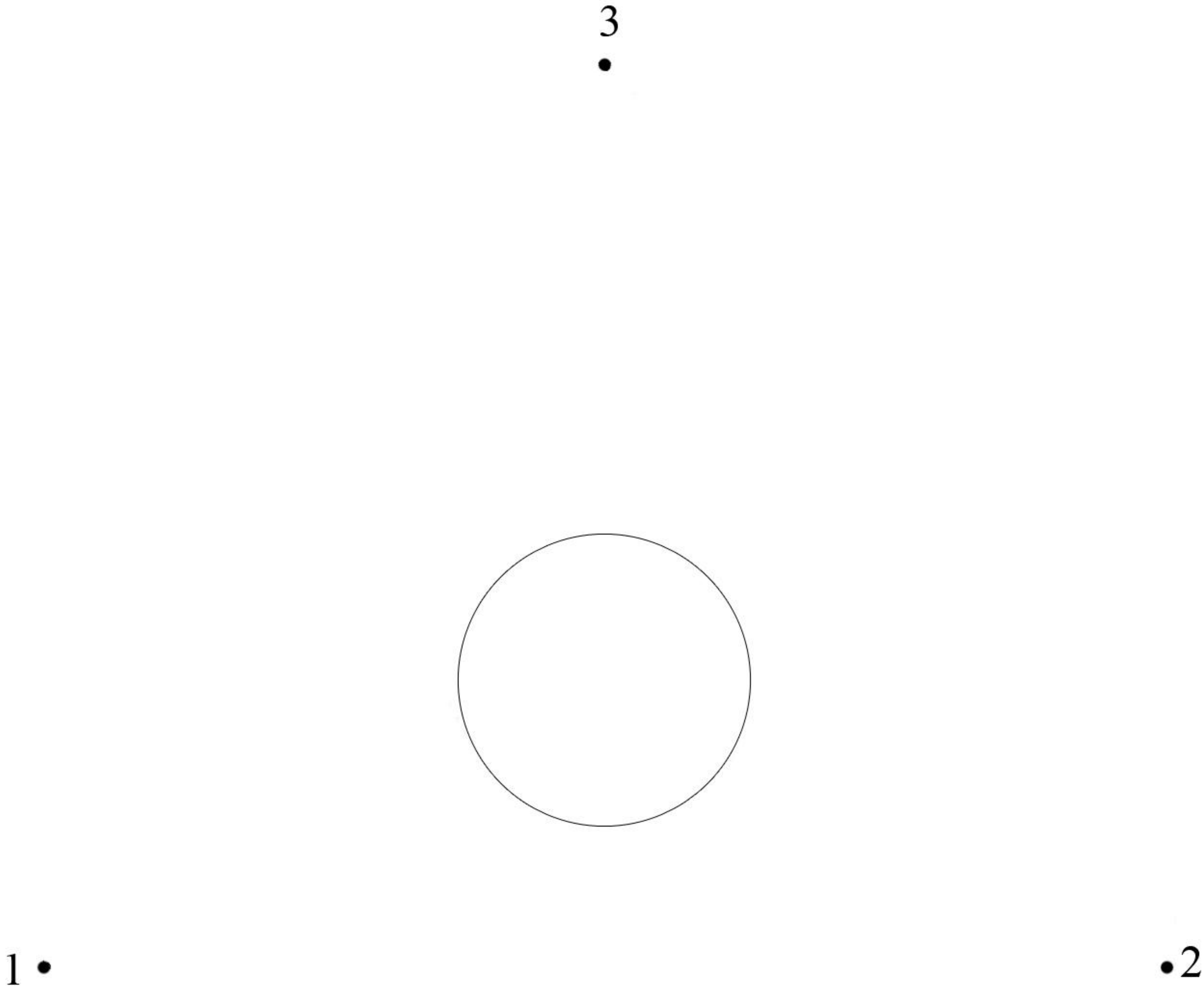


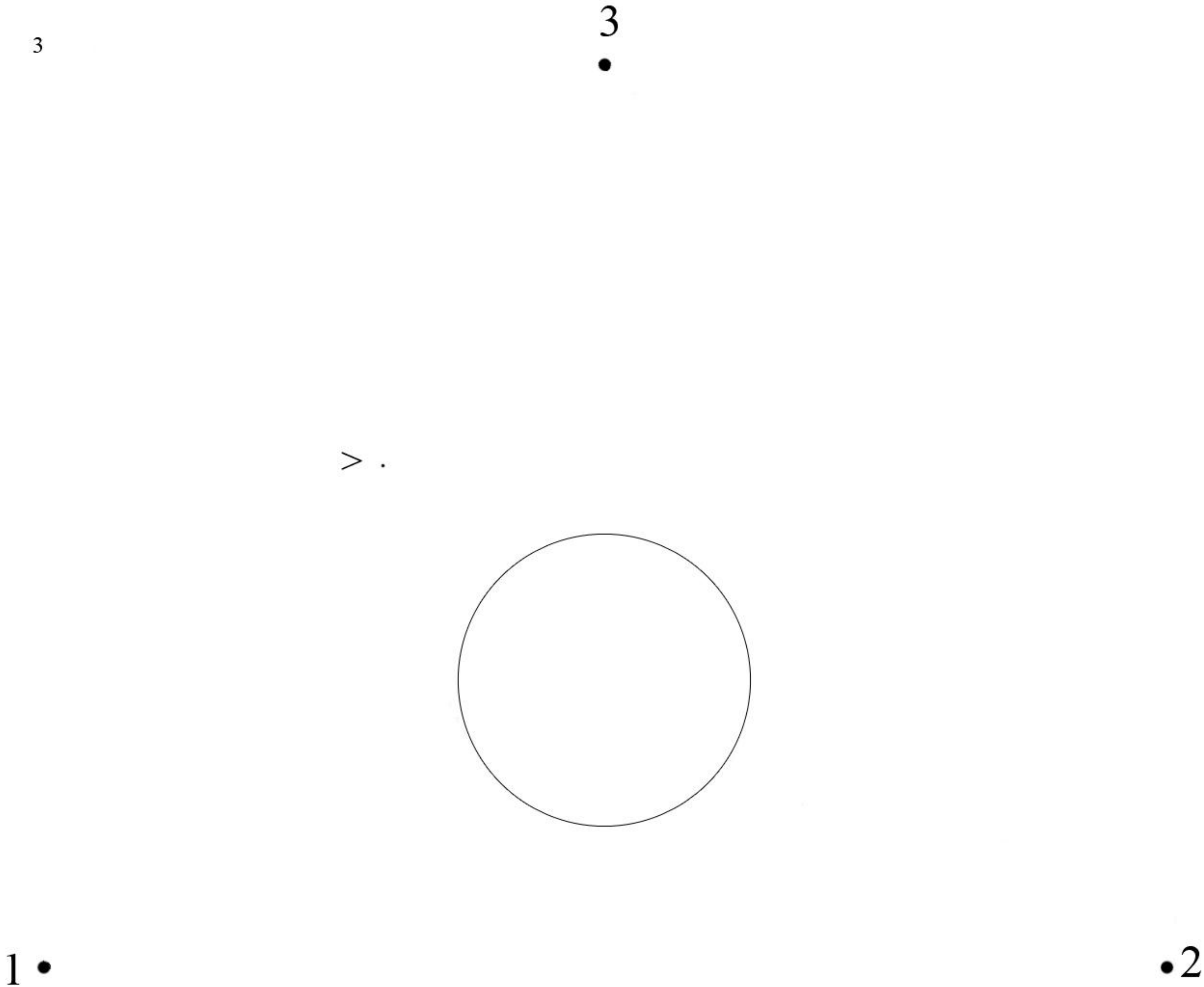
Is this a 'random walk'?

Far from it!

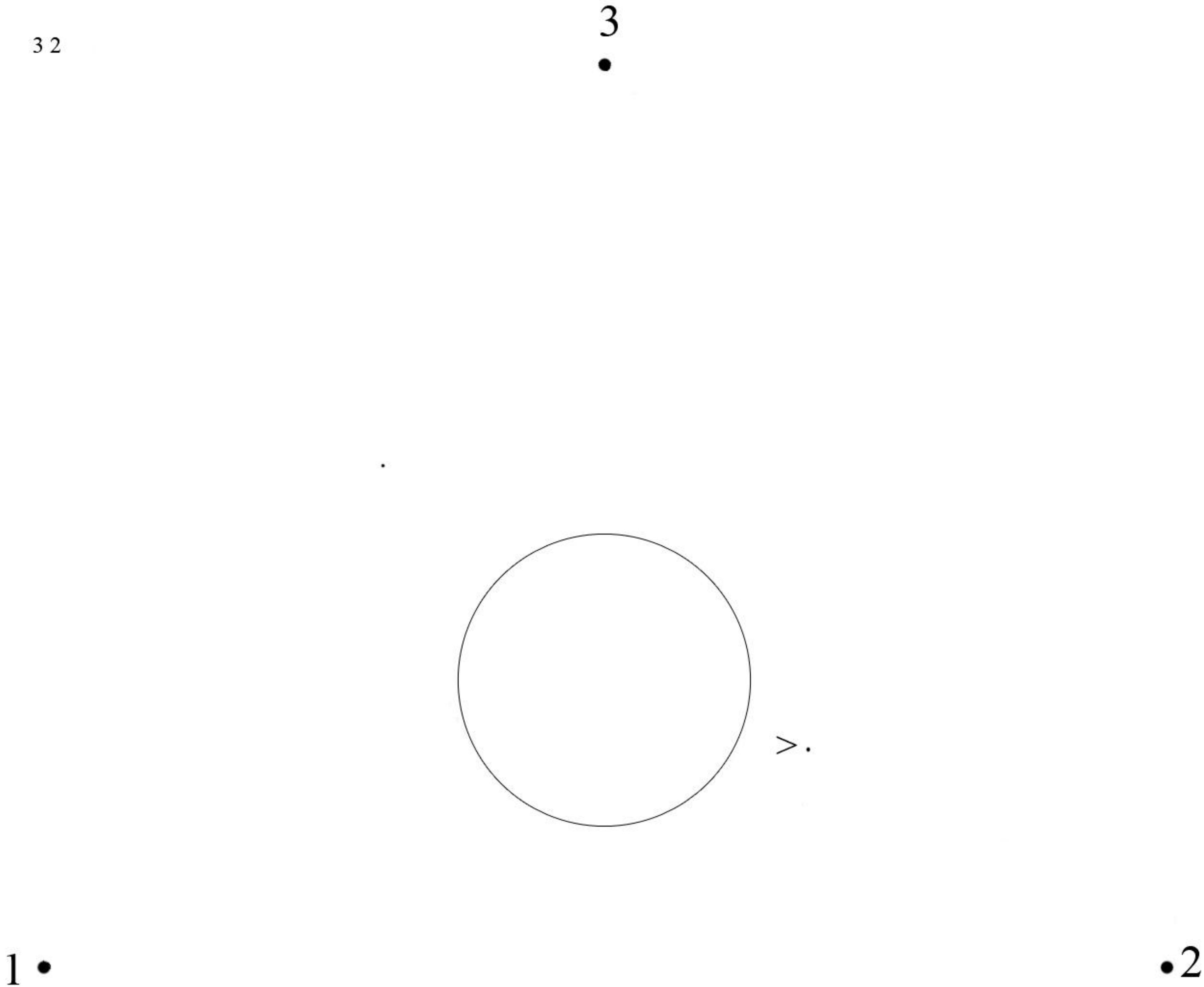
For example, a game point will <u>never</u> land in this circle

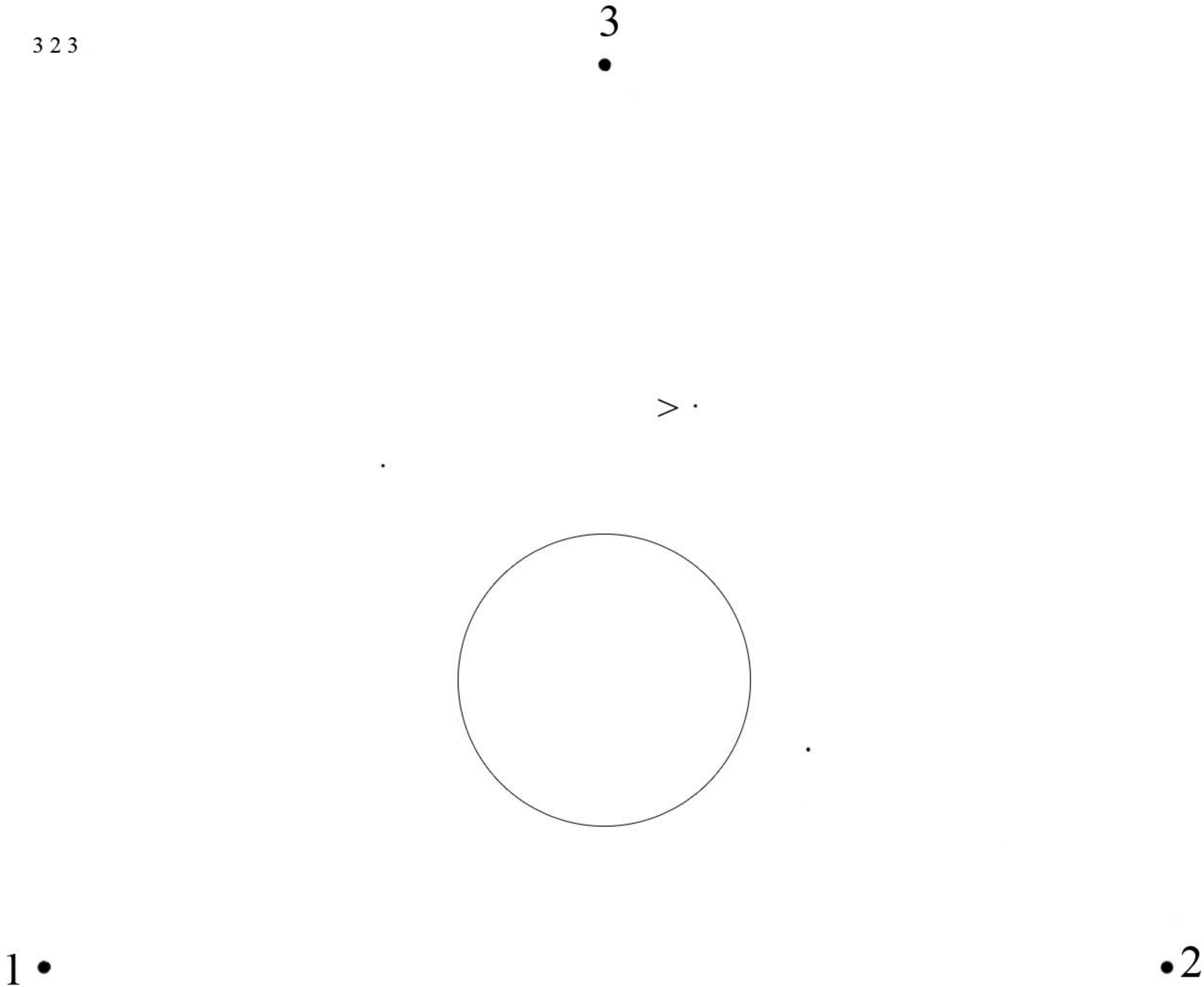


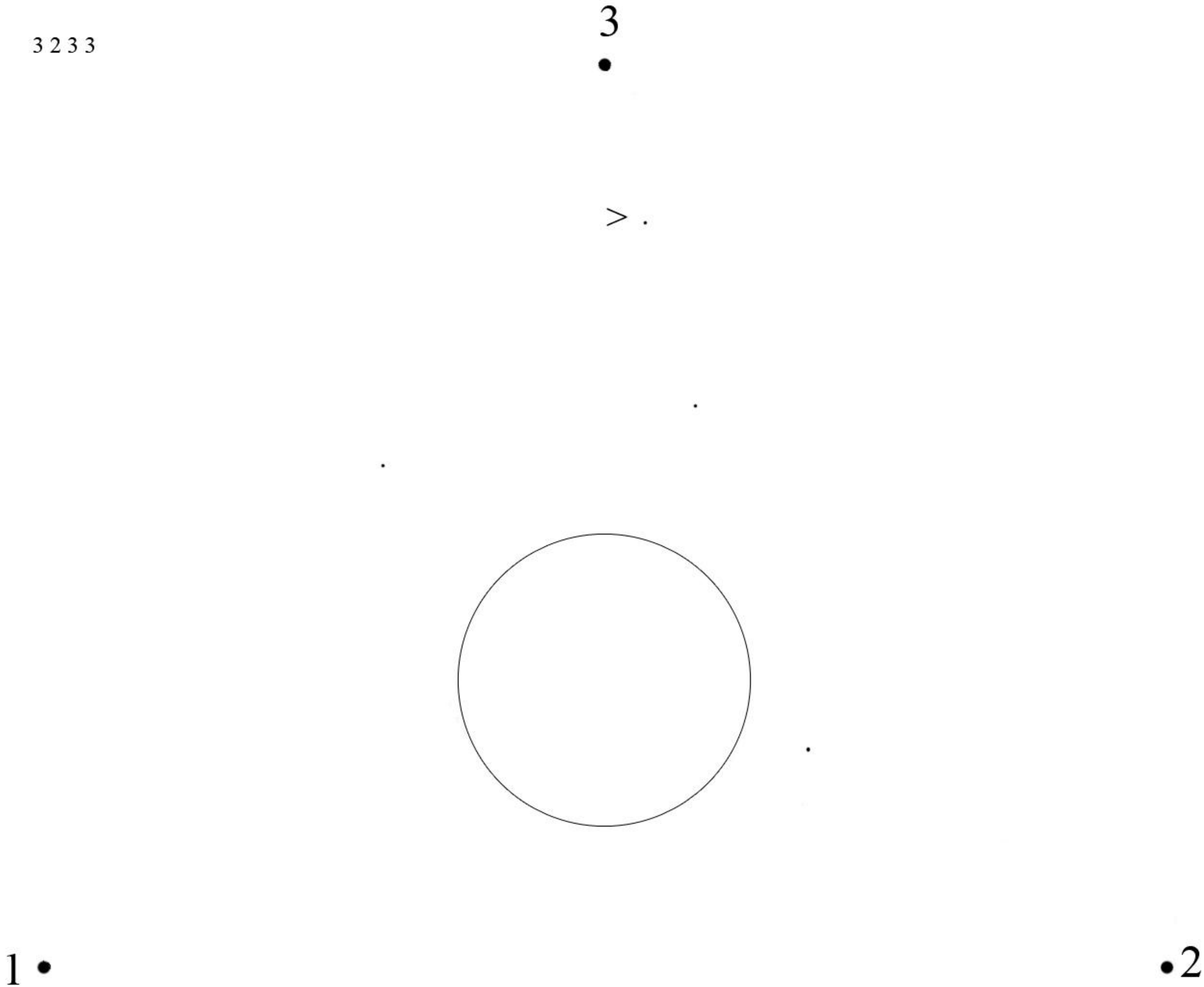

3
1
2


3
1
2

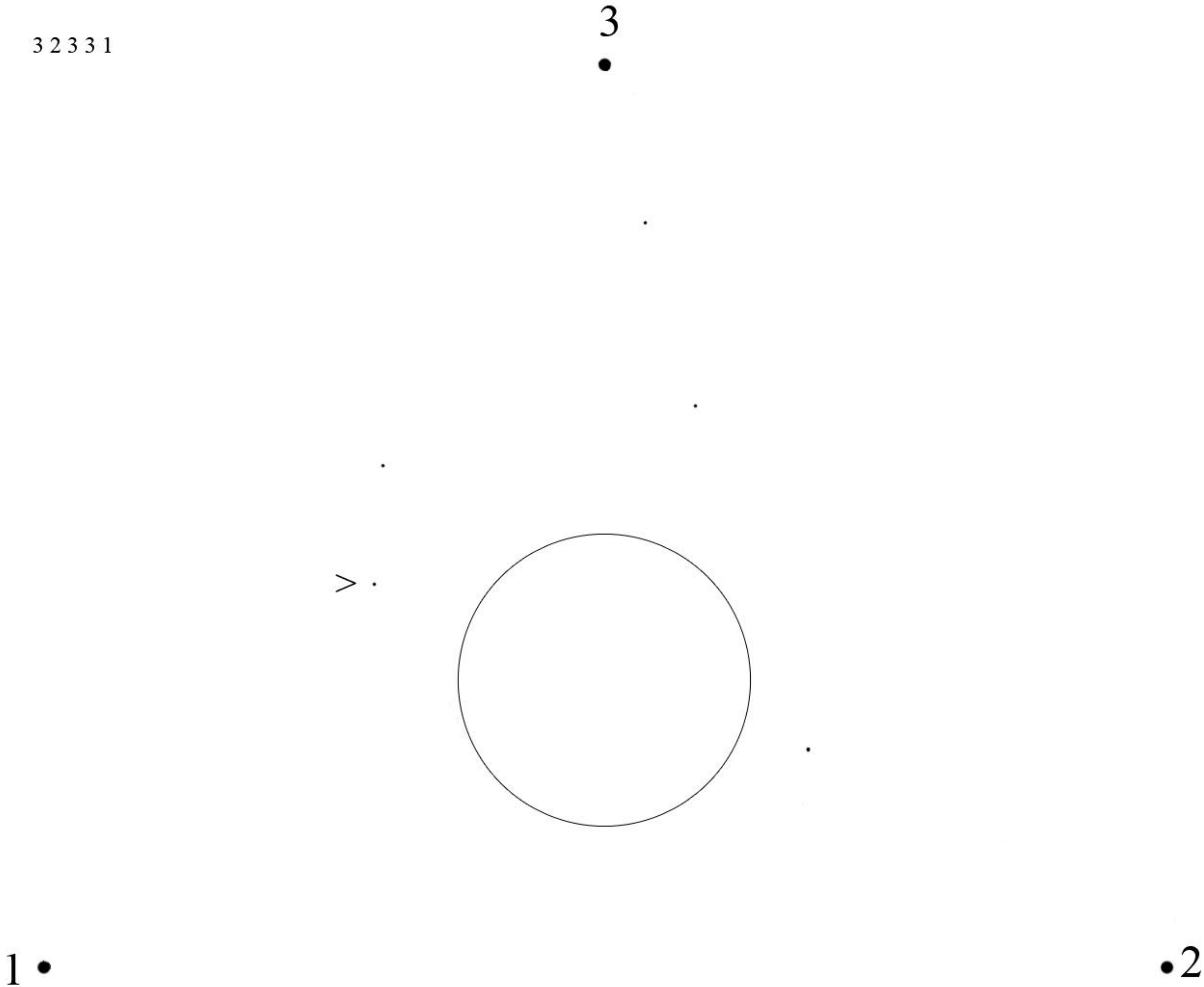
3 2



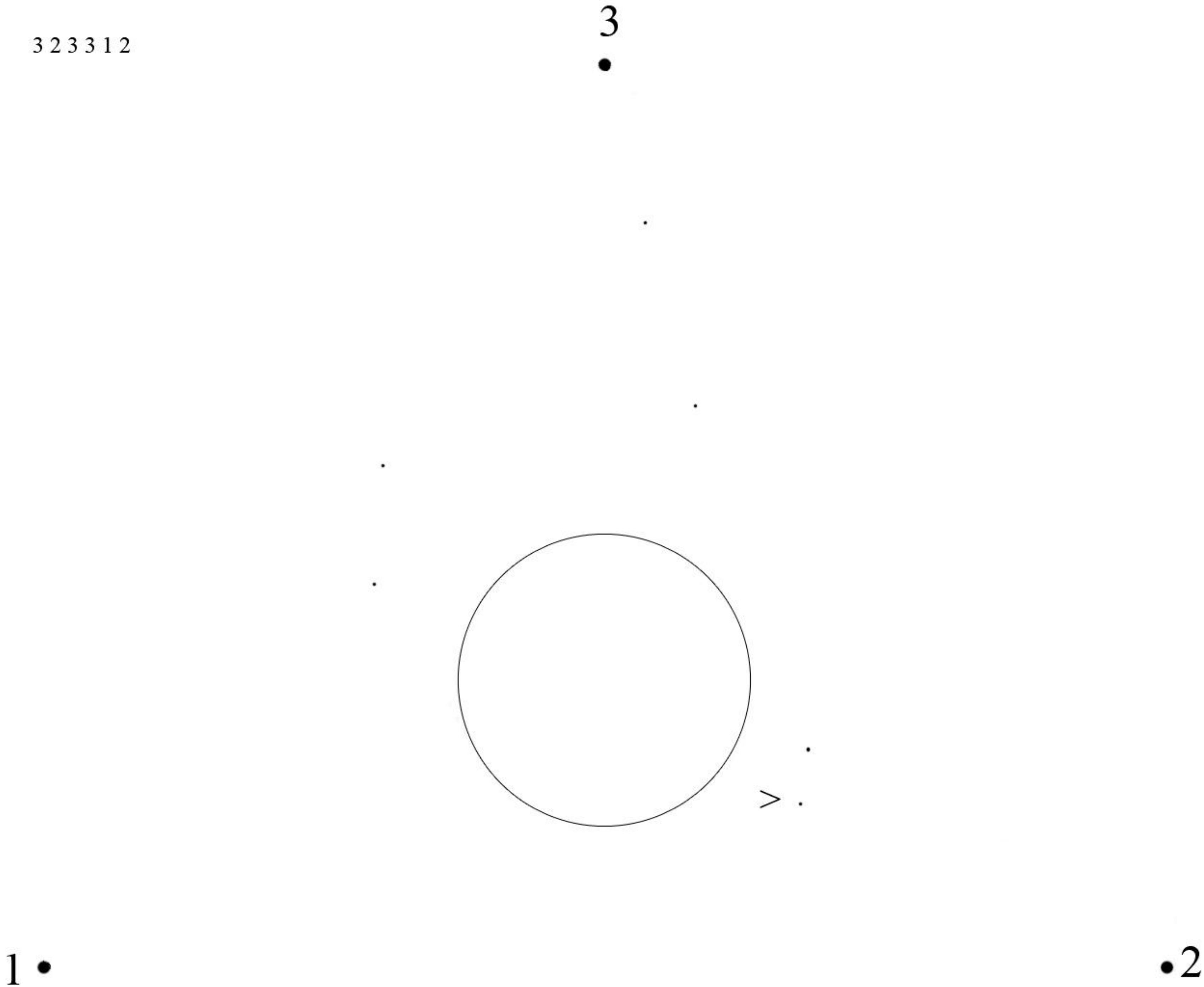

3 2 3
3
>
1
2


3 2 3 3
3
>
1
2

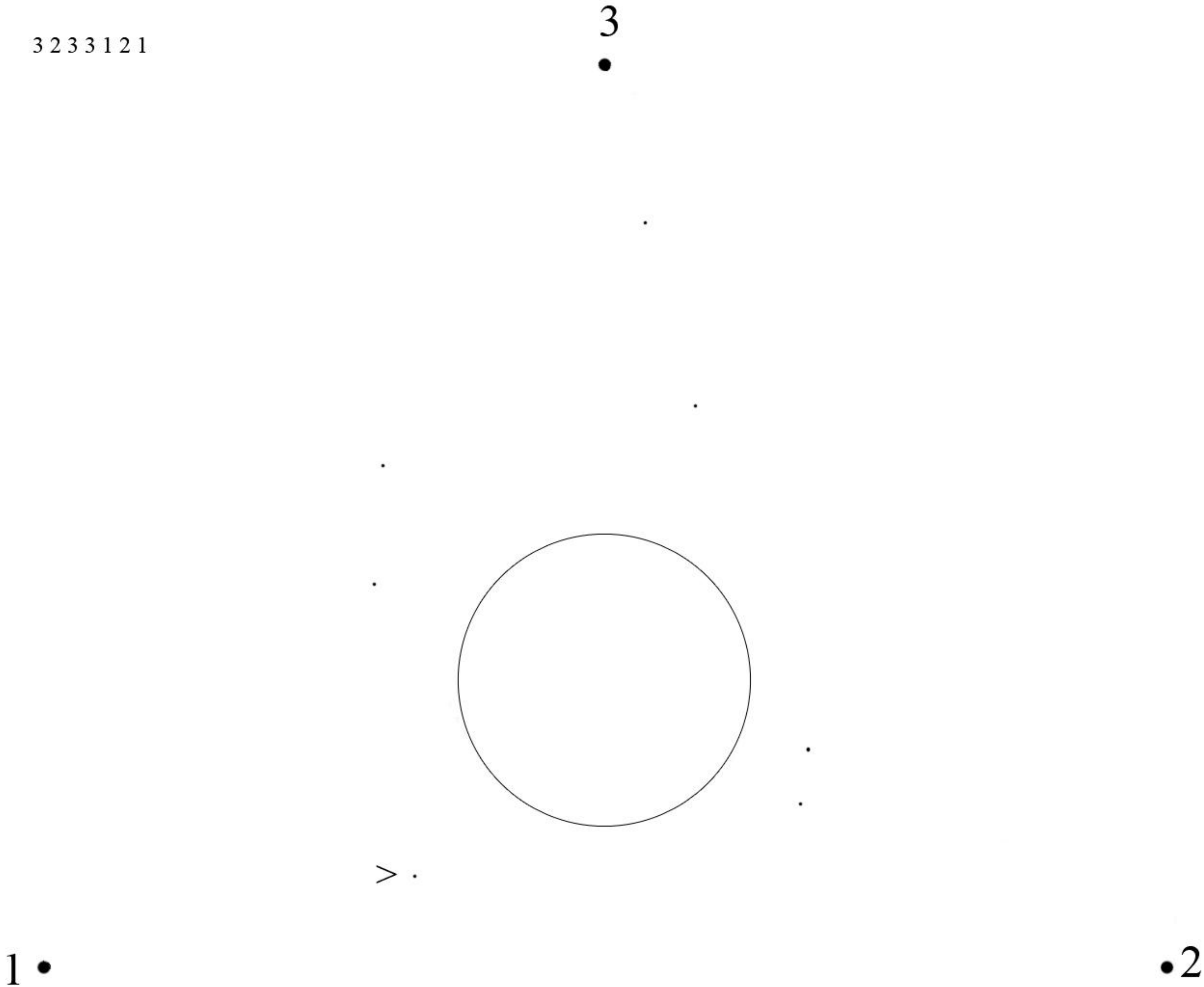
3 2 3 3 1

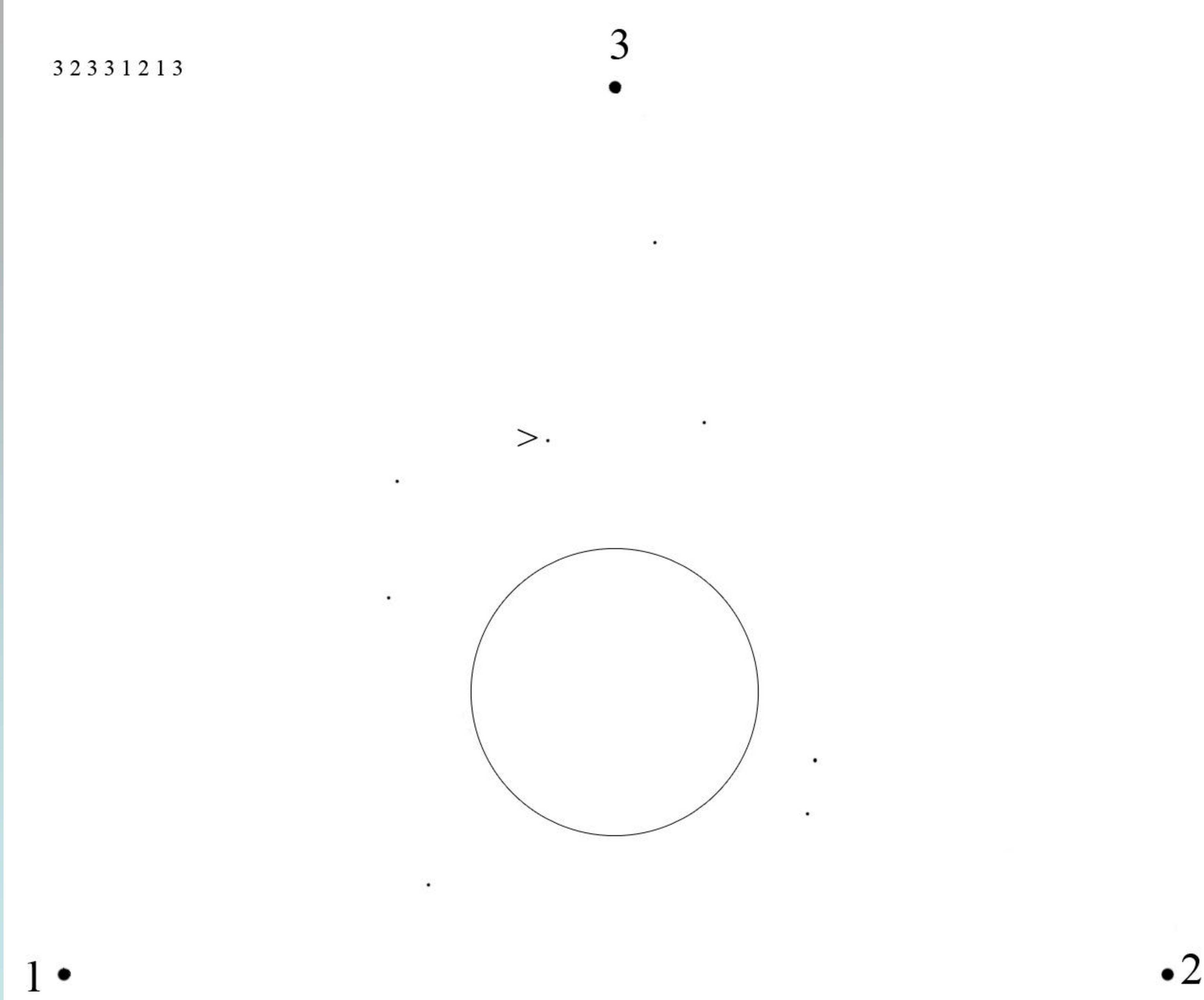


3 2 3 3 1 2

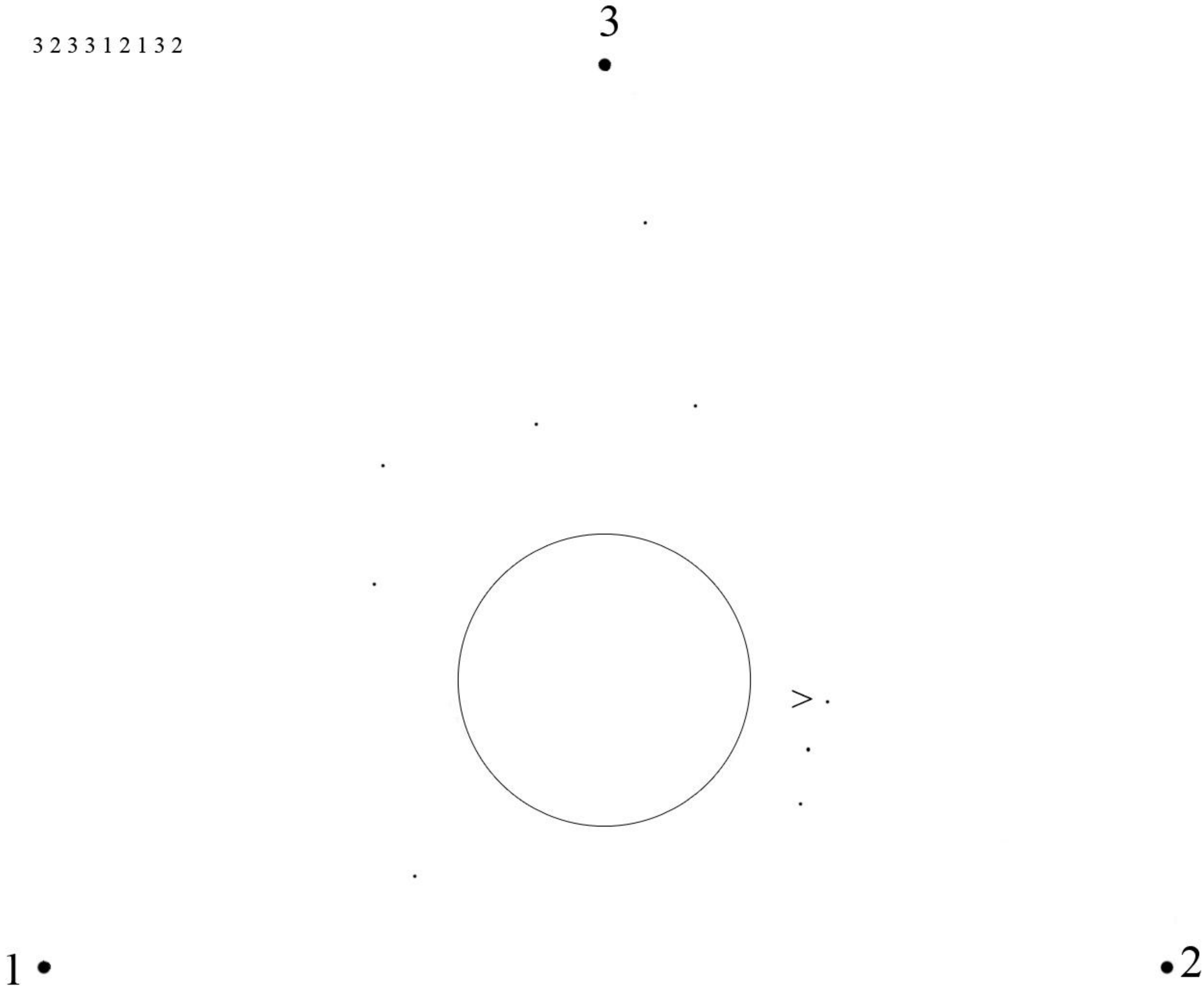


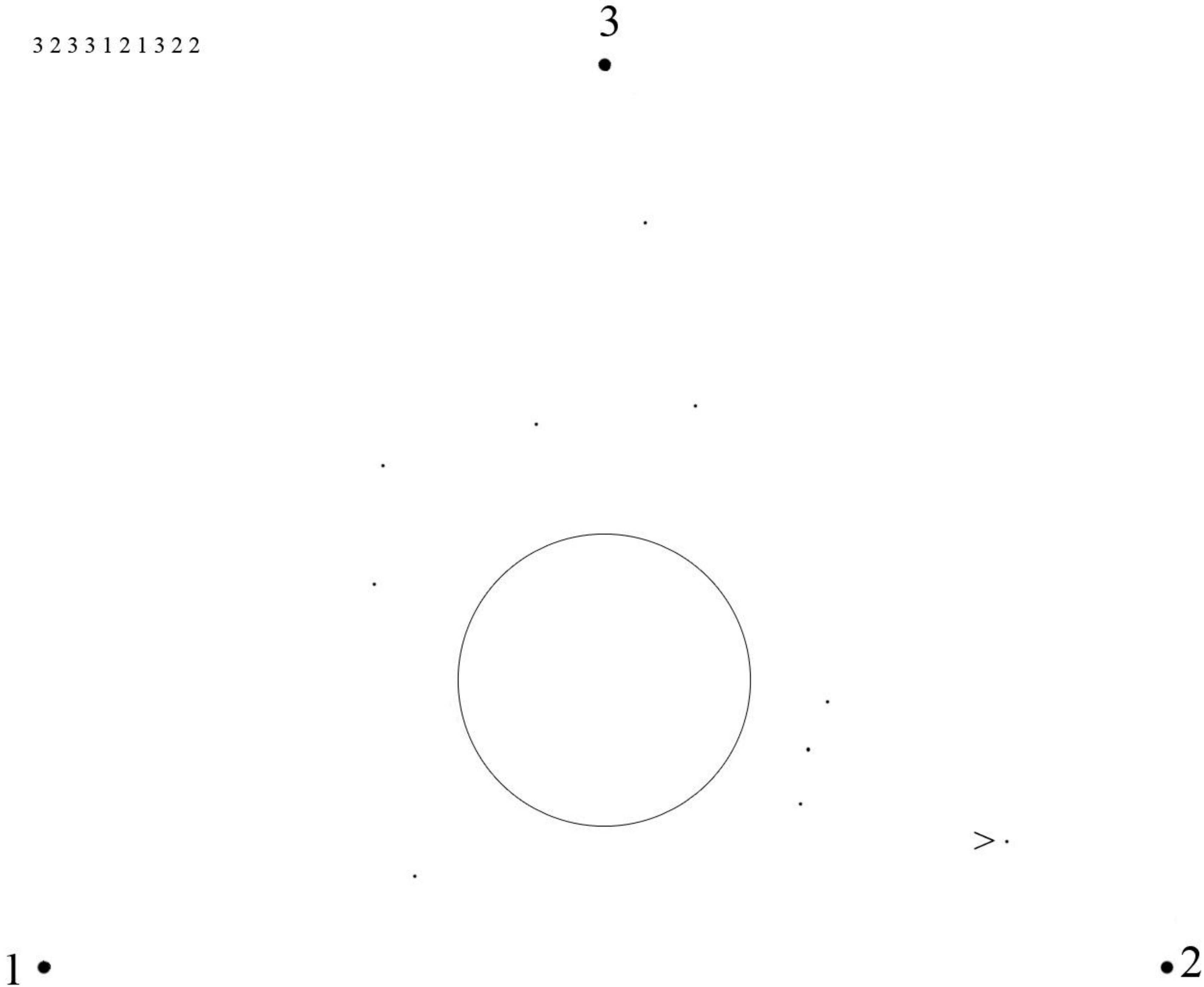
3 2 3 3 1 2 1

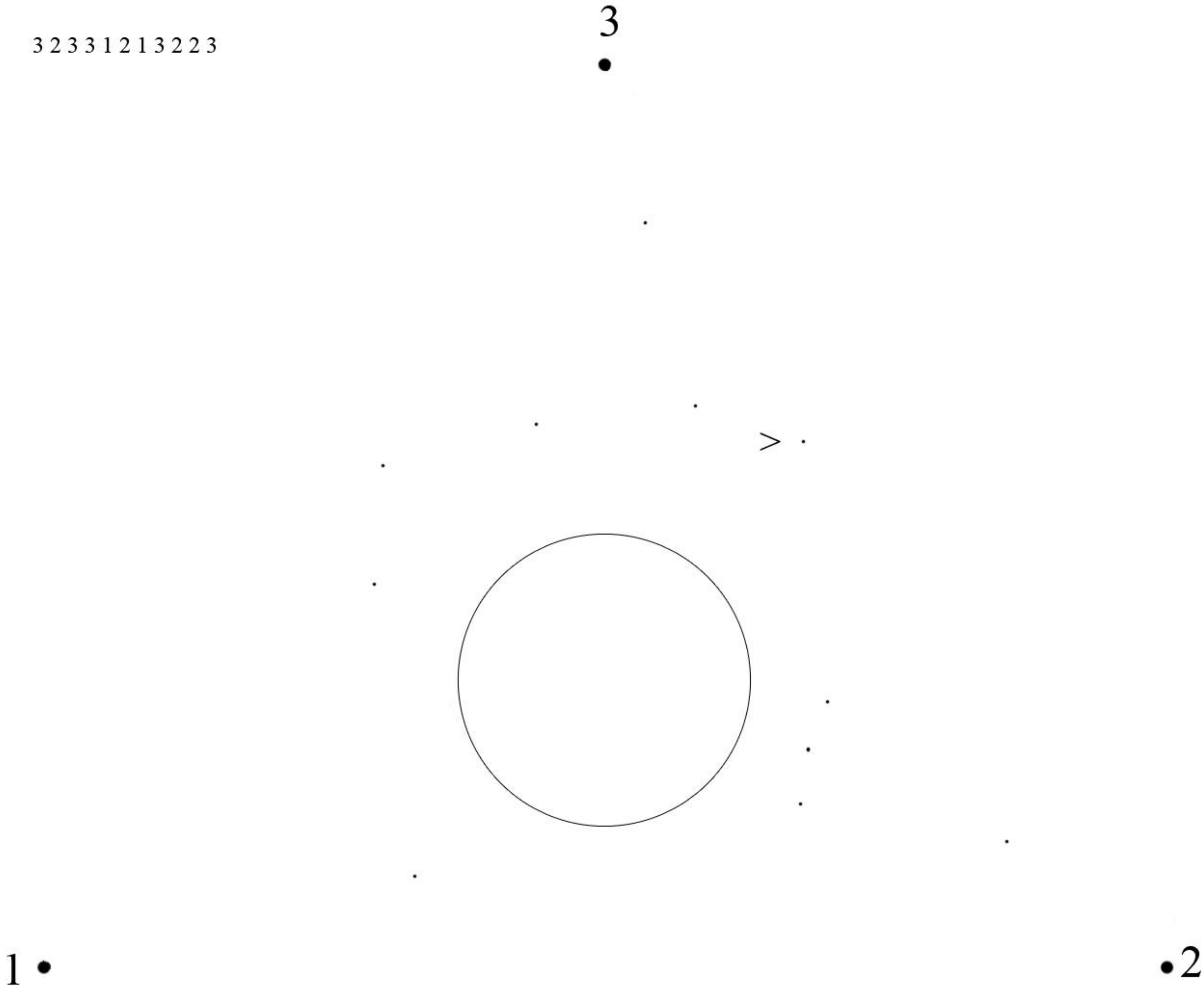


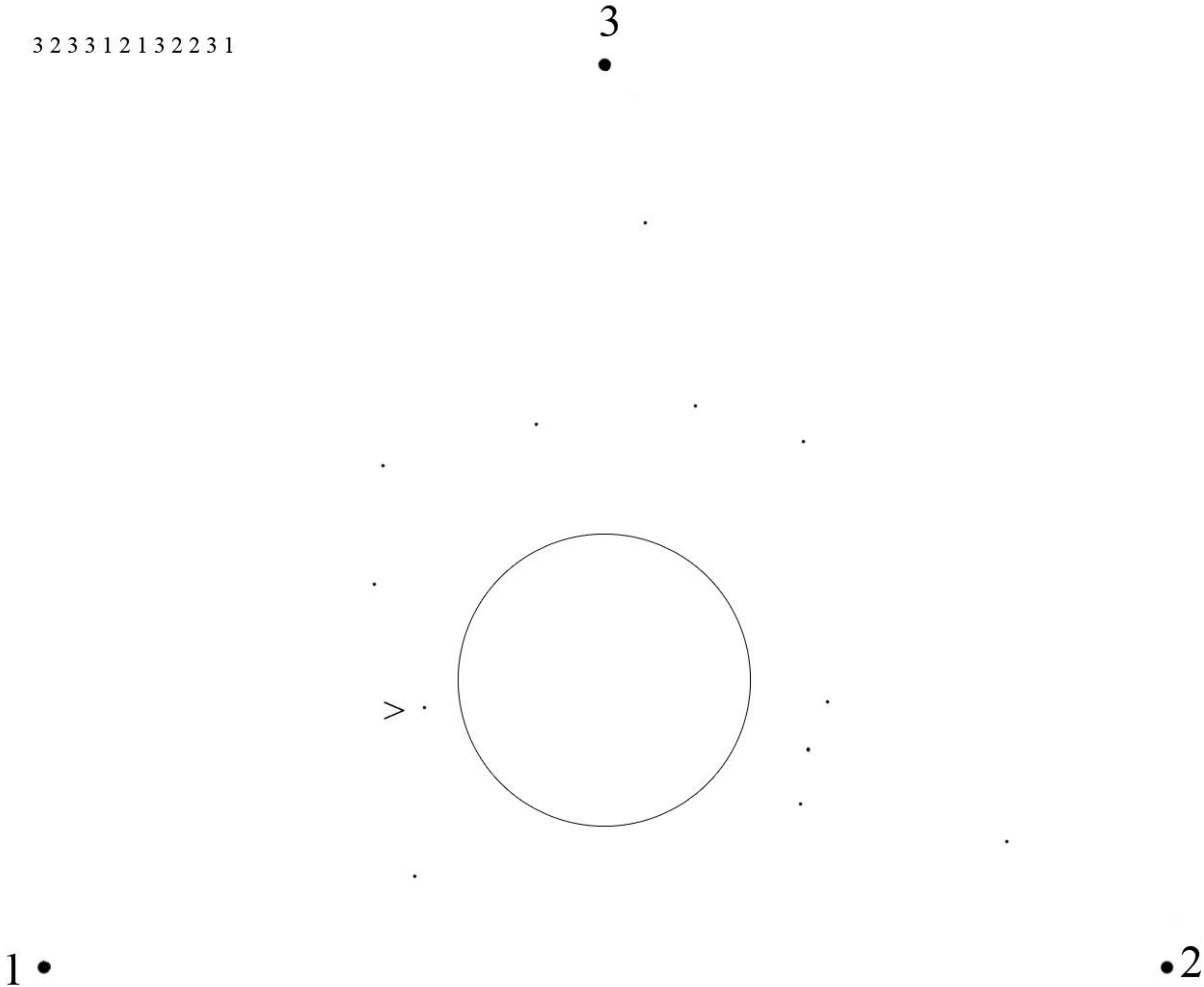

3 2 3 3 1 2 1 3
3
1
2

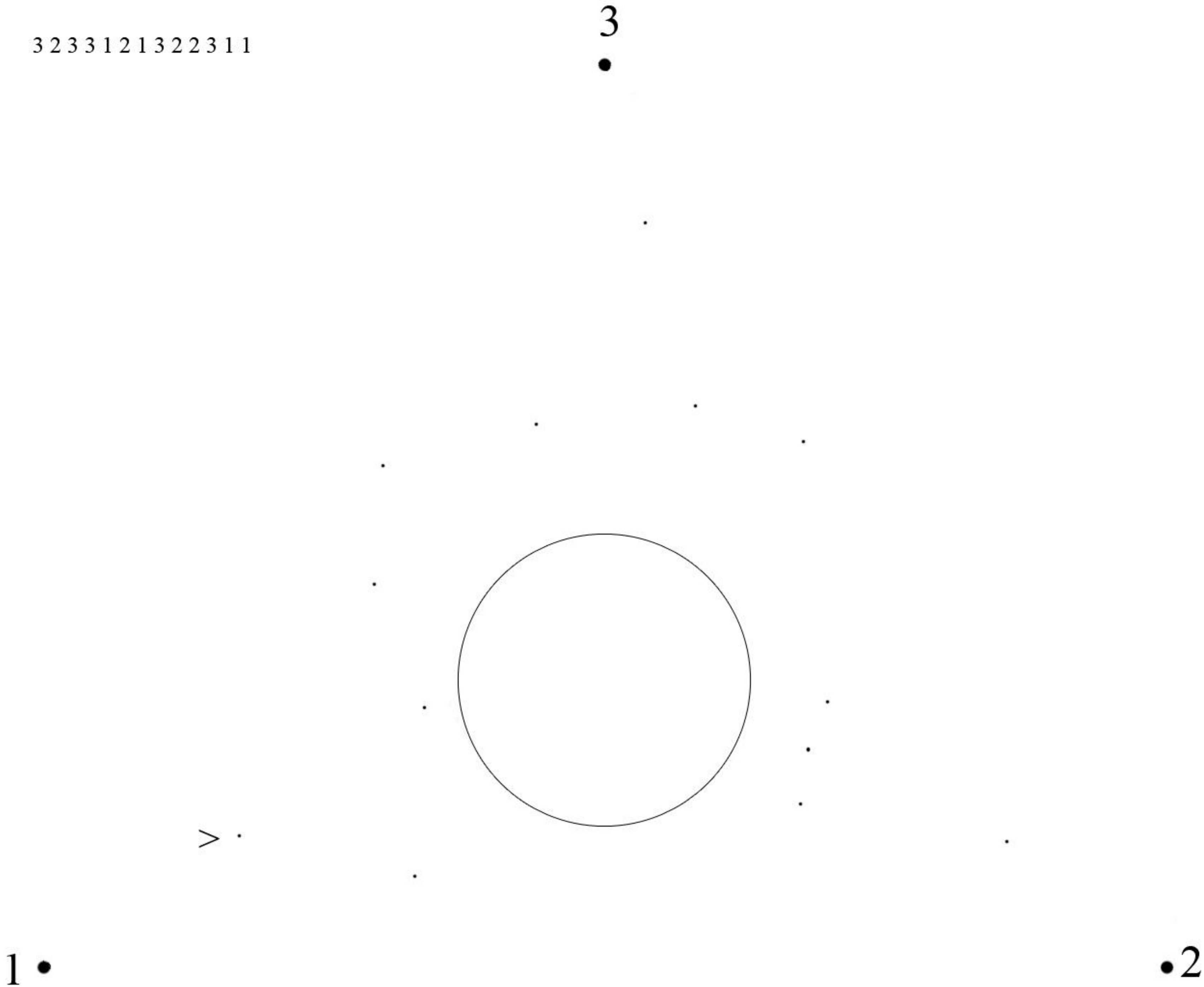
3 2 3 3 1 2 1 3 2

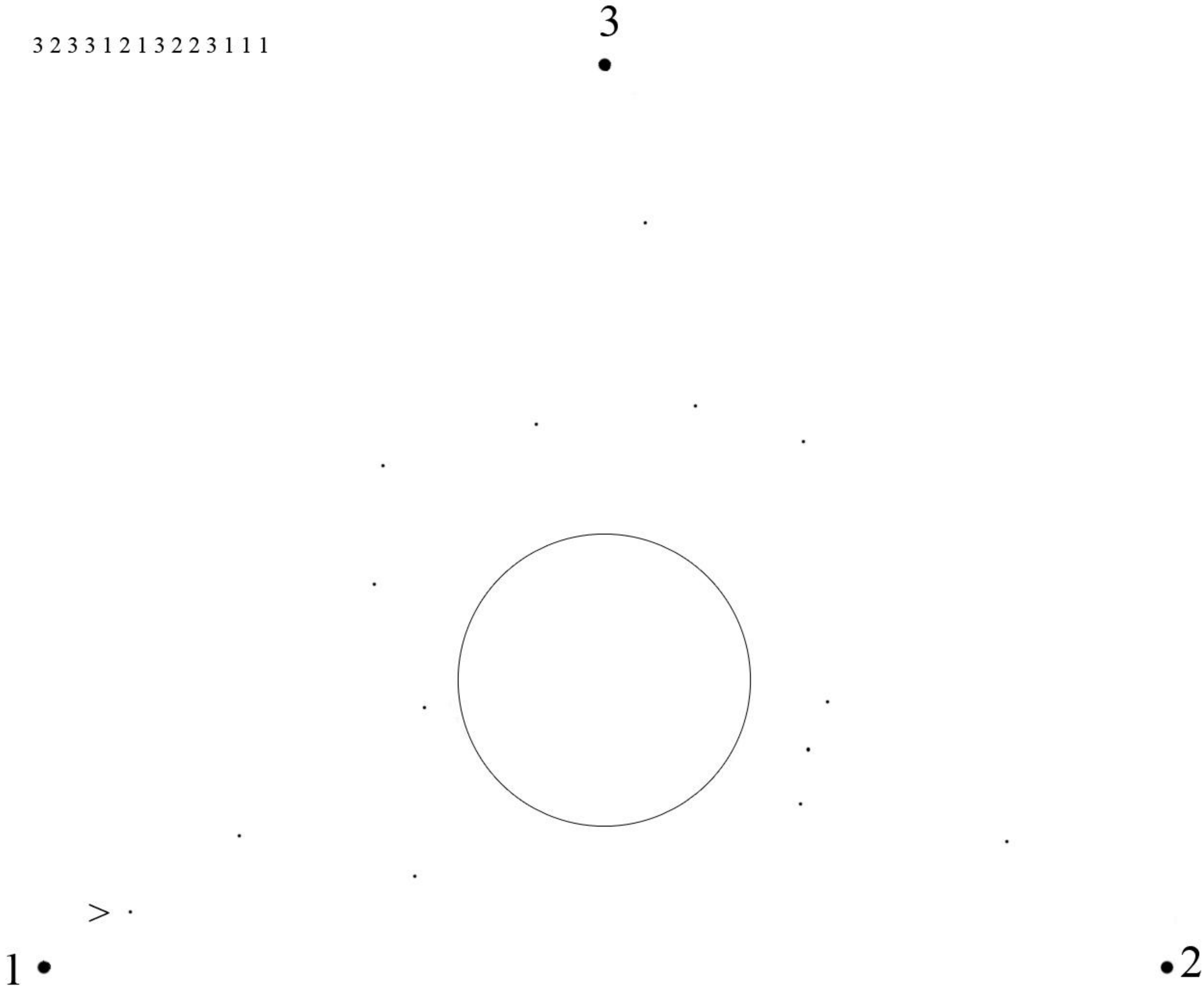


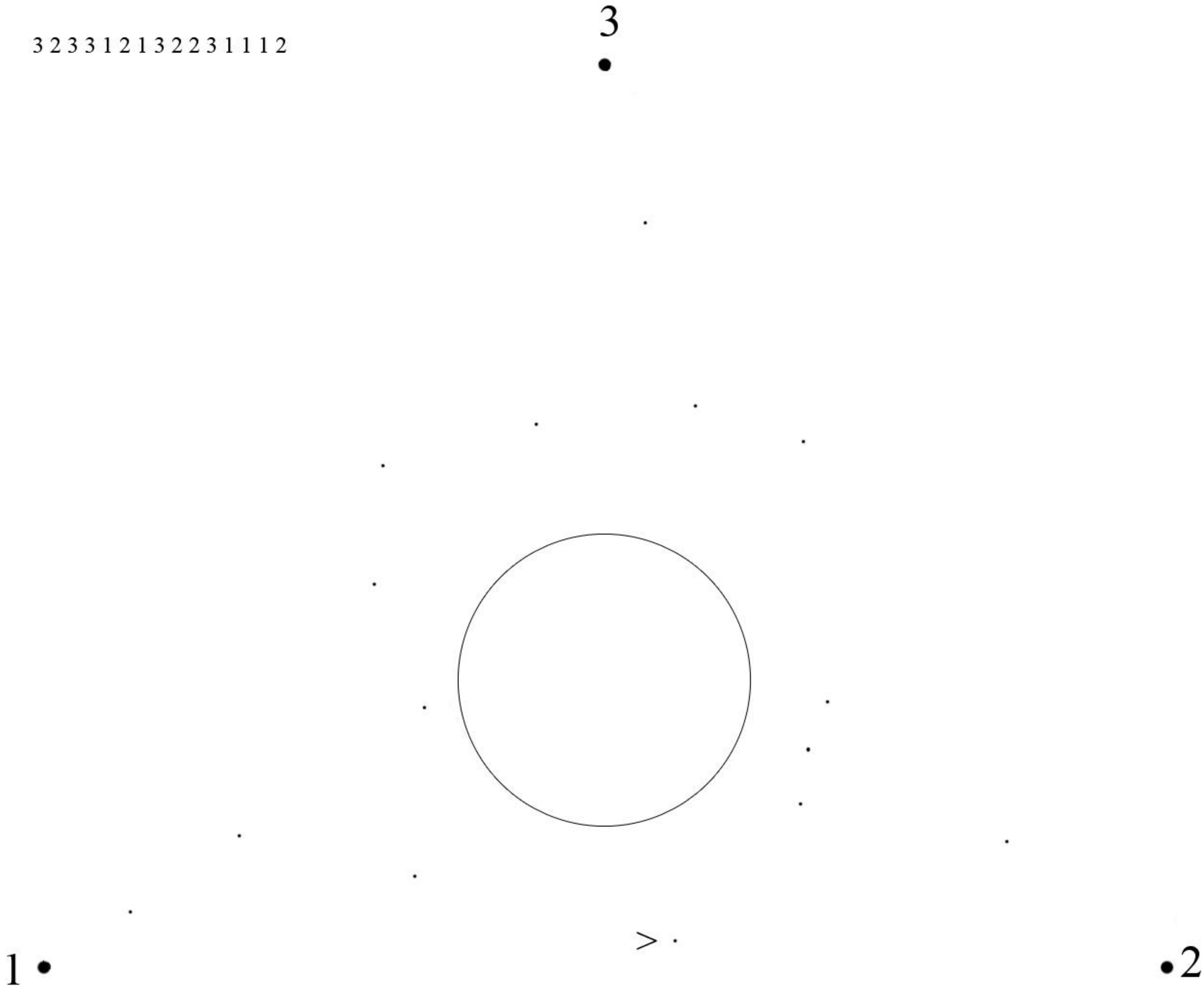

3 2 3 3 1 2 1 3 2 2
3
>
1
2

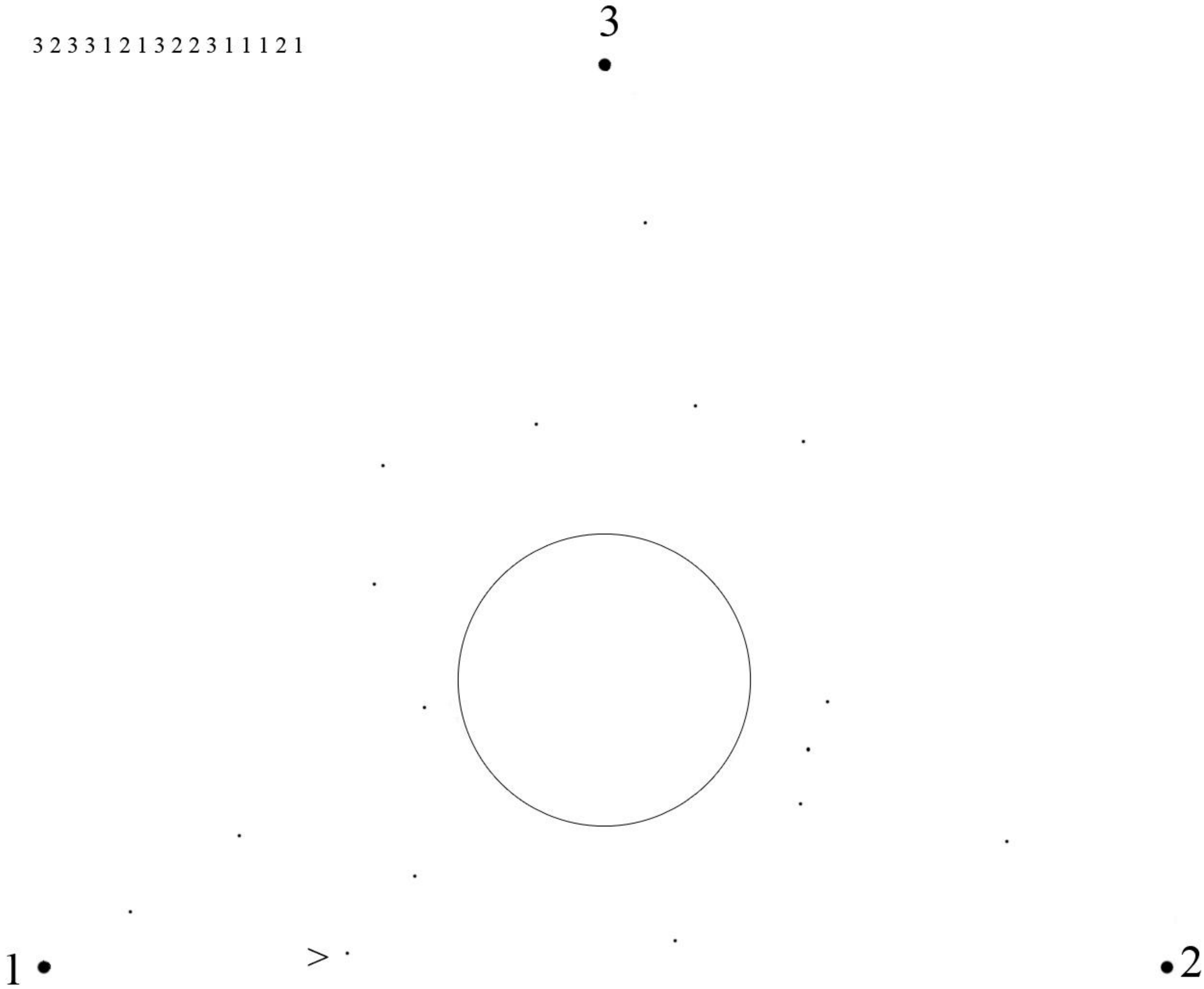

3 2 3 3 1 2 1 3 2 2 3
3
1
2
>


3 2 3 3 1 2 1 3 2 2 3 1
3
1
2
>

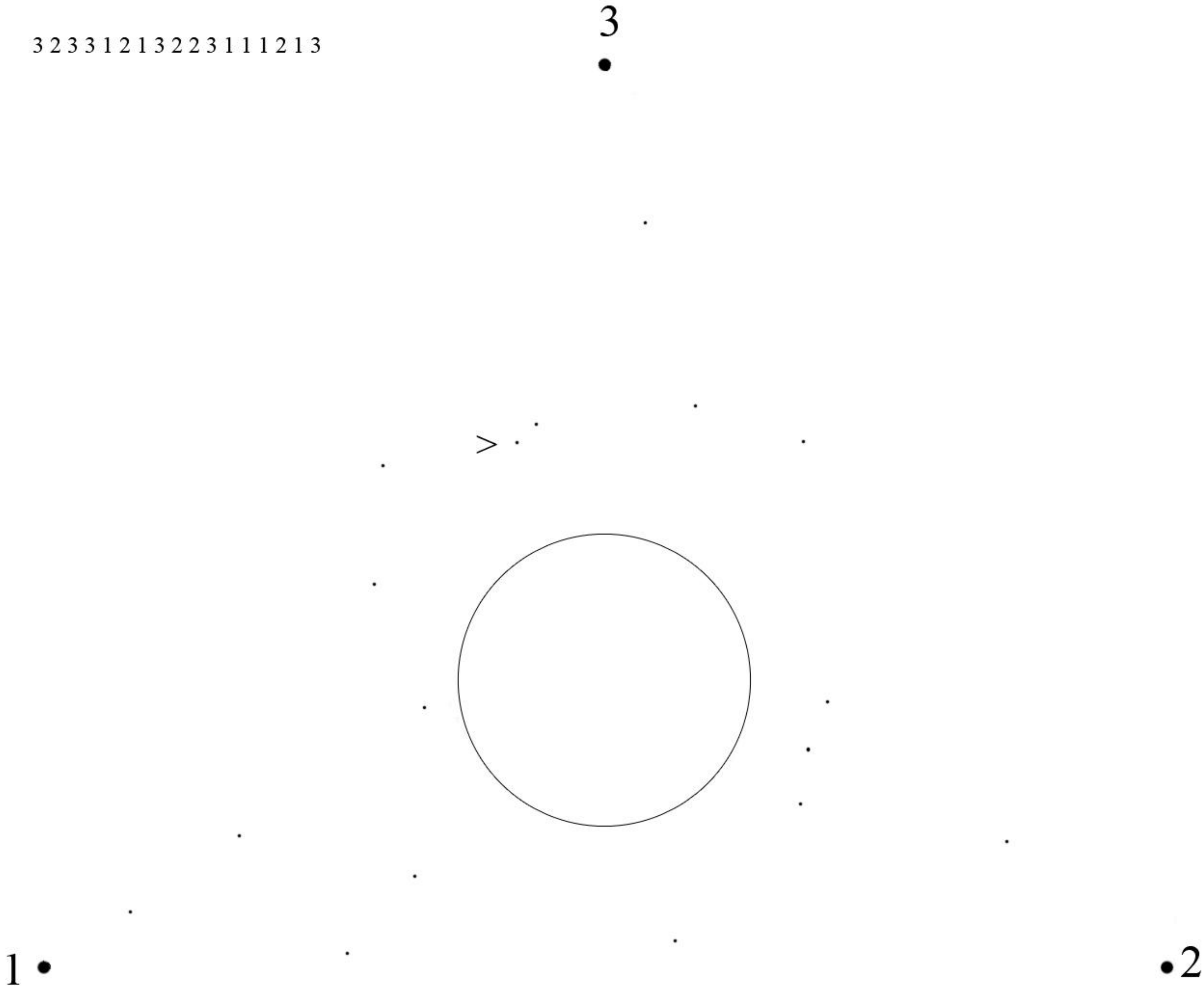

3 2 3 3 1 2 1 3 2 2 3 1 1
3
>
1
2


3 2 3 3 1 2 1 3 2 2 3 1 1 1
3
1
2
>

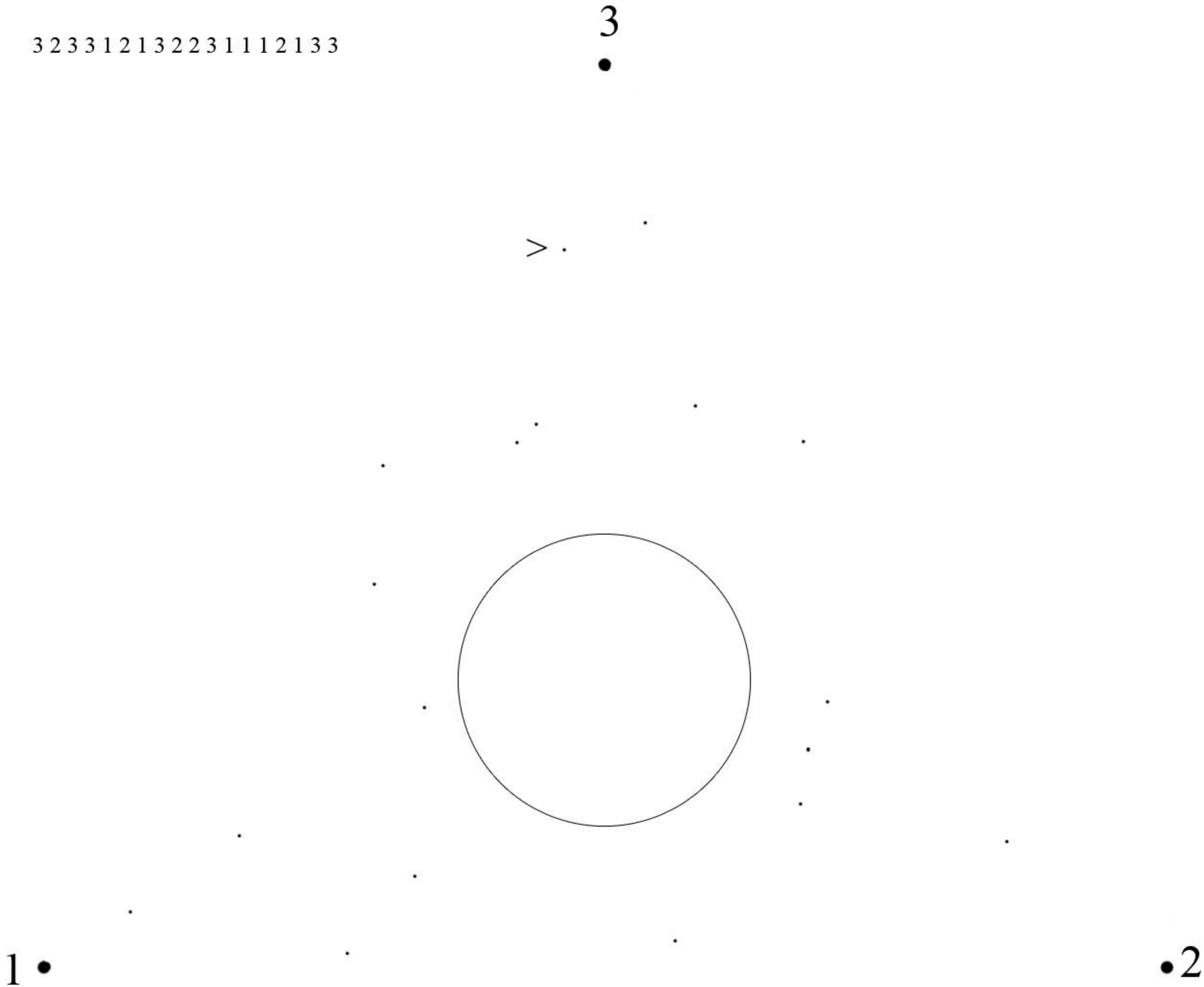

3 2 3 3 1 2 1 3 2 2 3 1 1 1 2
3
1
2
>

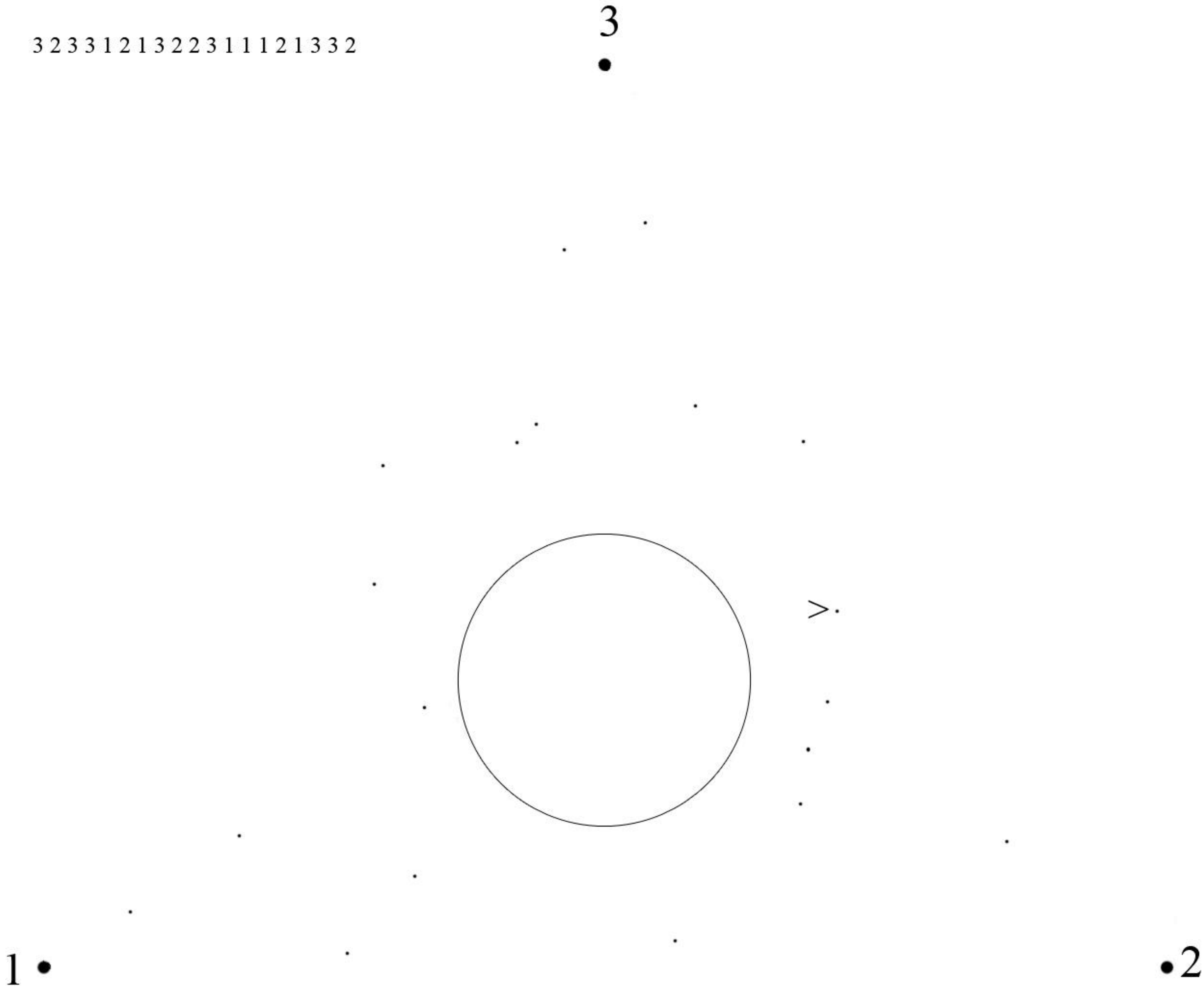

3 2 3 3 1 2 1 3 2 2 3 1 1 1 2 1
3
1
>
2

3 2 3 3 1 2 1 3 2 2 3 1 1 1 2 1 3

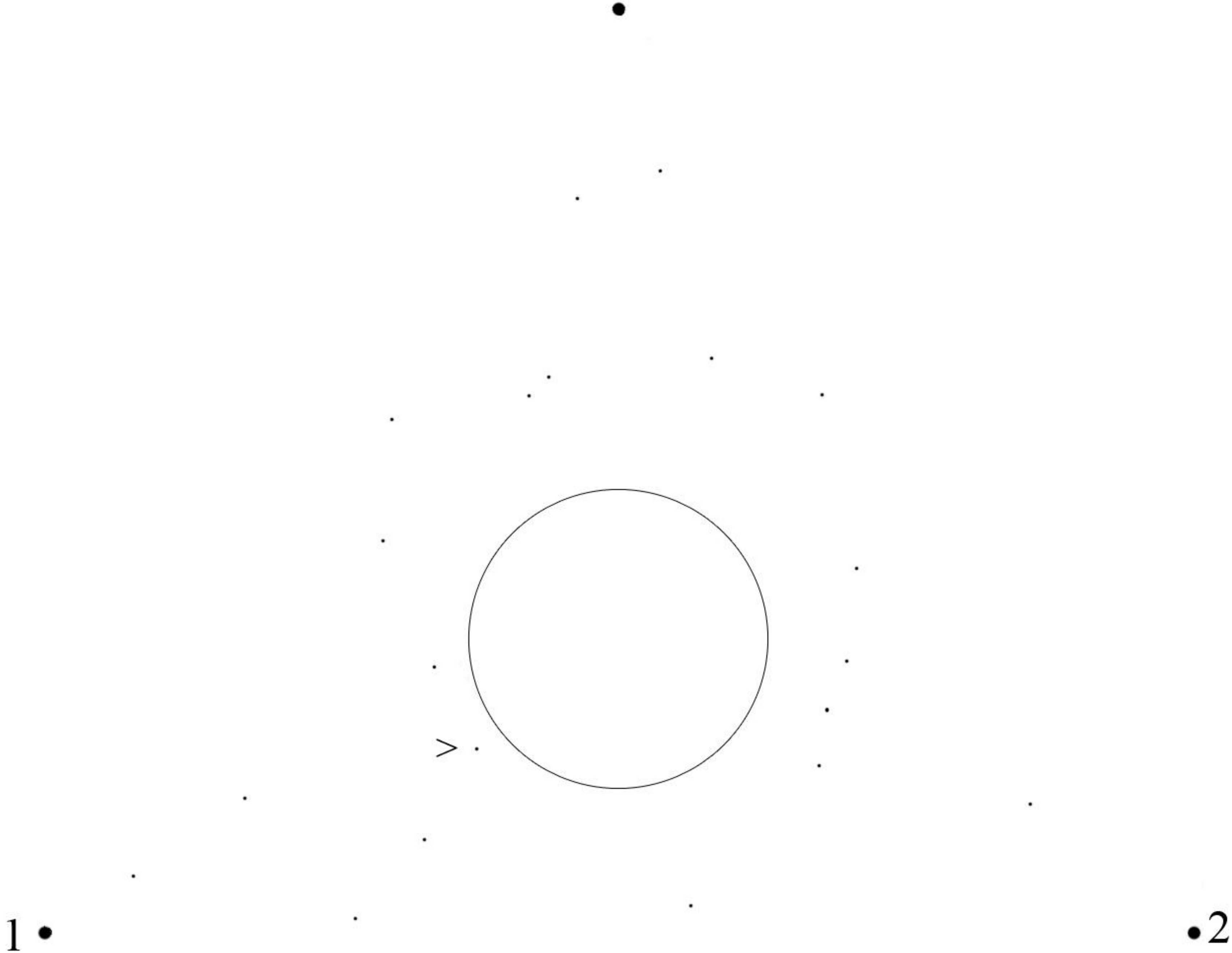


3 2 3 3 1 2 1 3 2 2 3 1 1 1 2 1 3 3

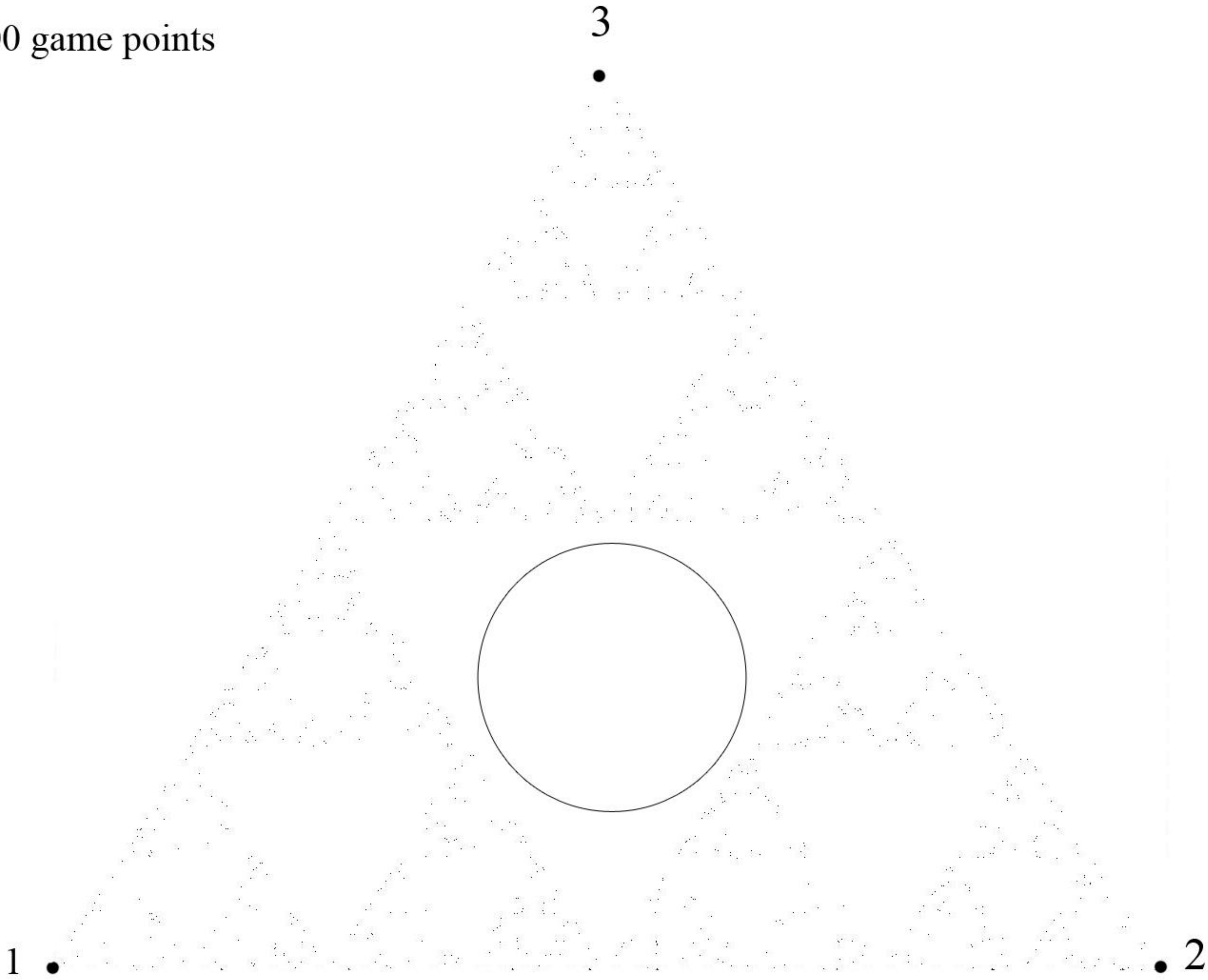


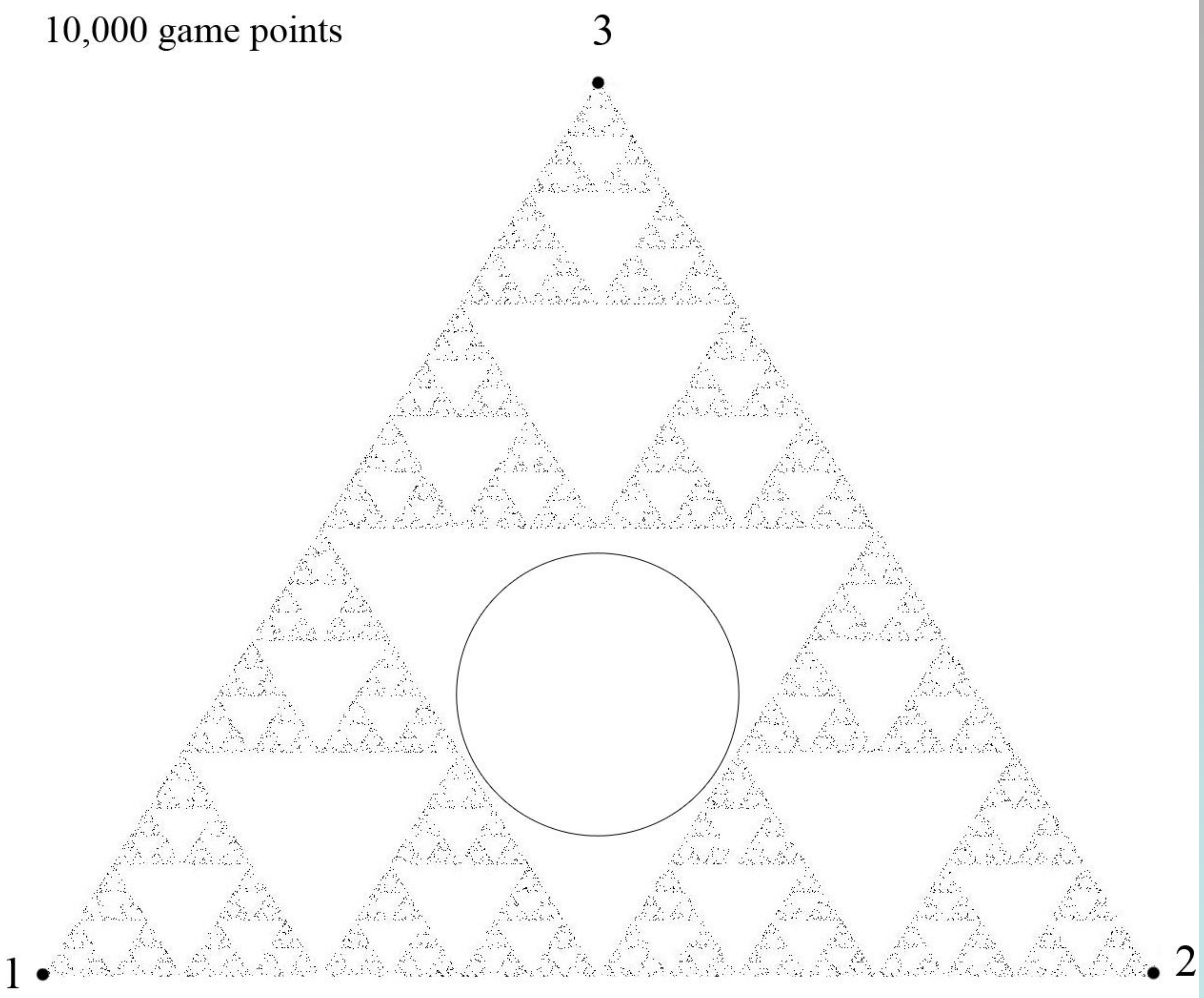

3 2 3 3 1 2 1 3 2 2 3 1 1 1 2 1 3 3 2
3
>
1
2

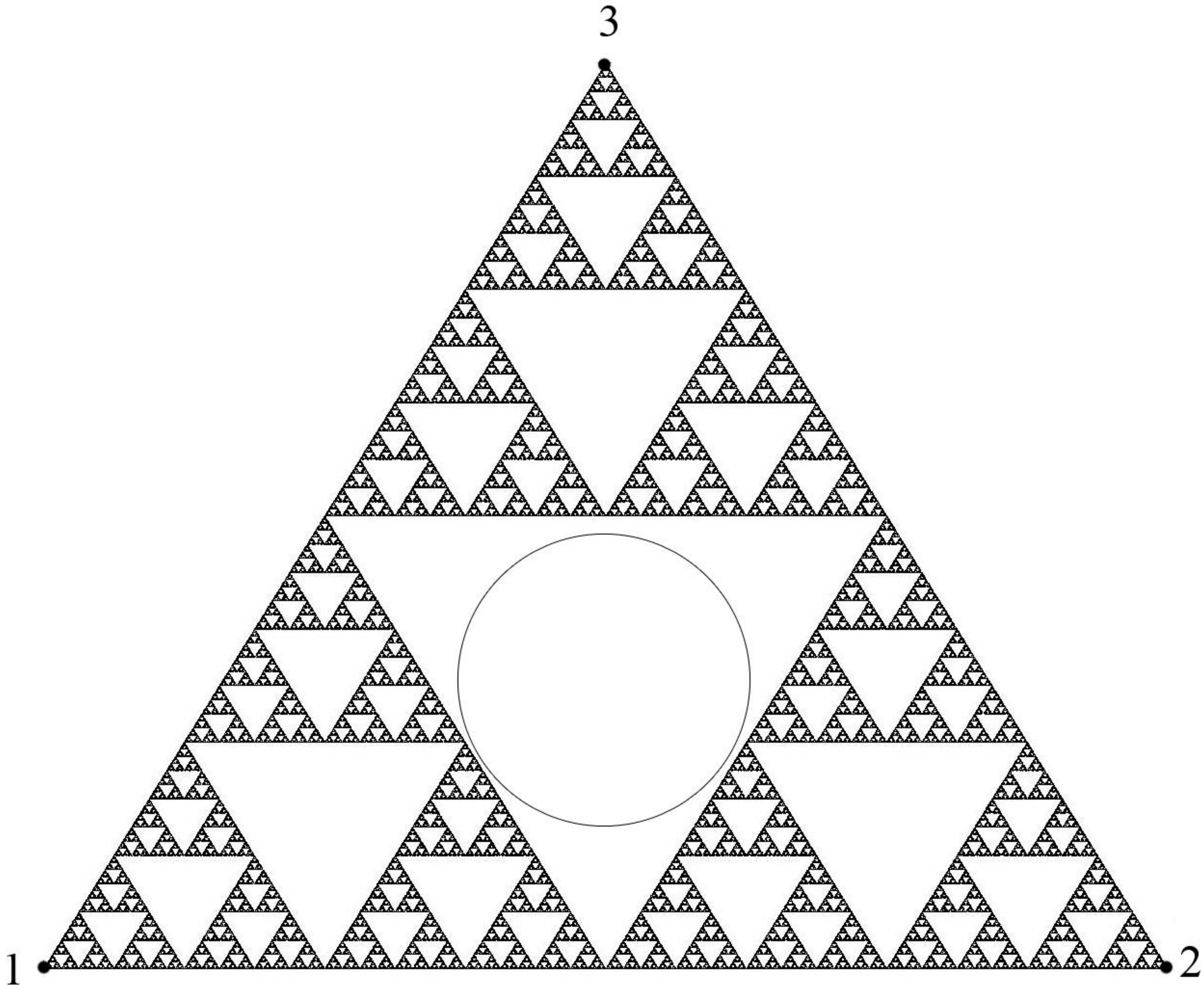
3 2 3 3 1 2 1 3 2 2 3 1 1 1 2 1 3 3 2 1



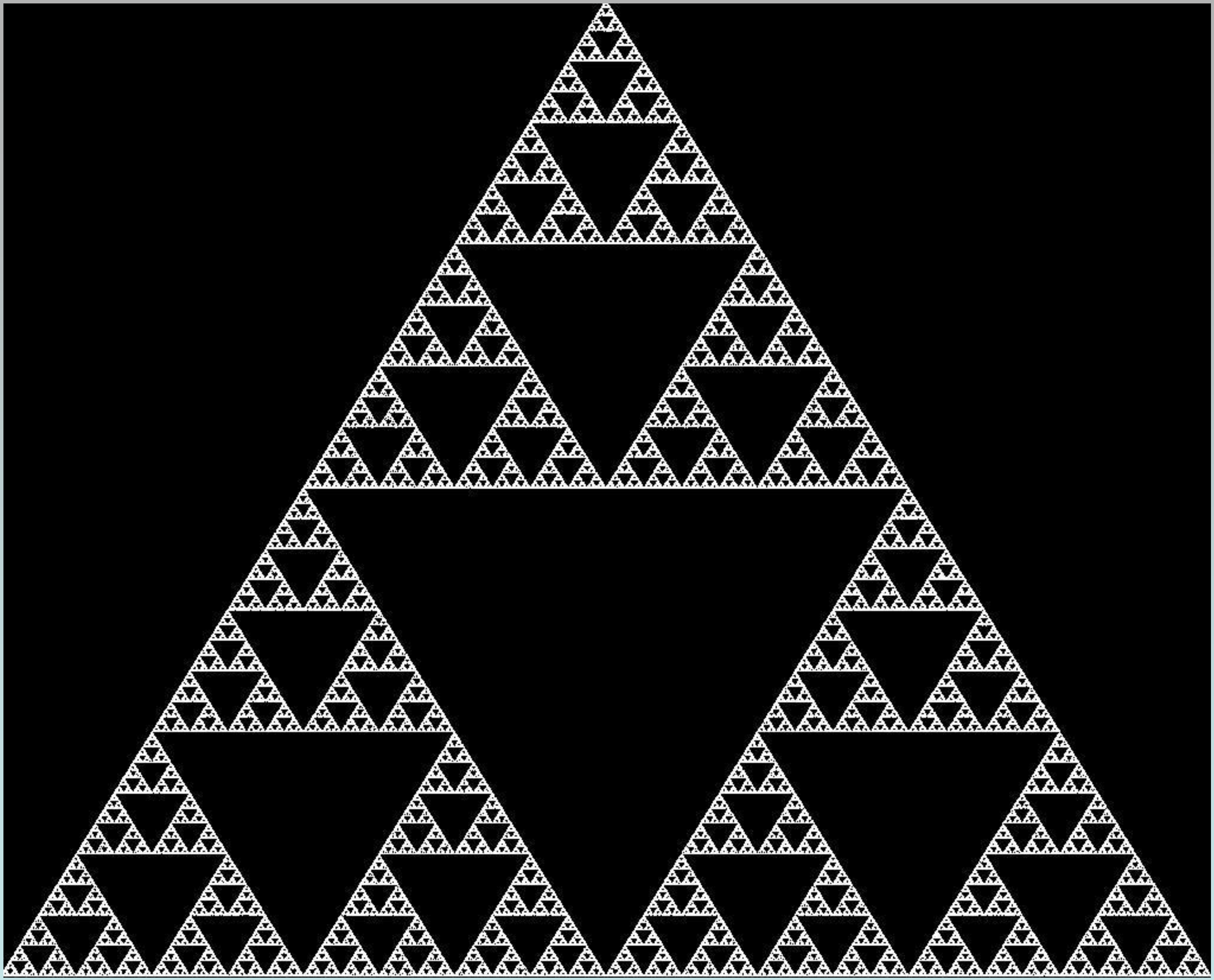
1000 game points




10,000 game points
3
1
2

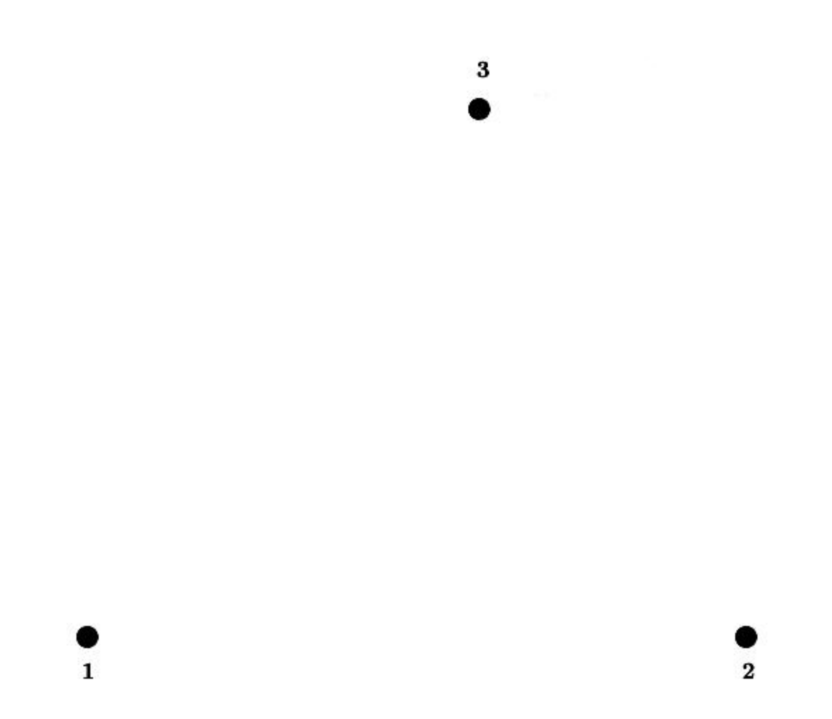


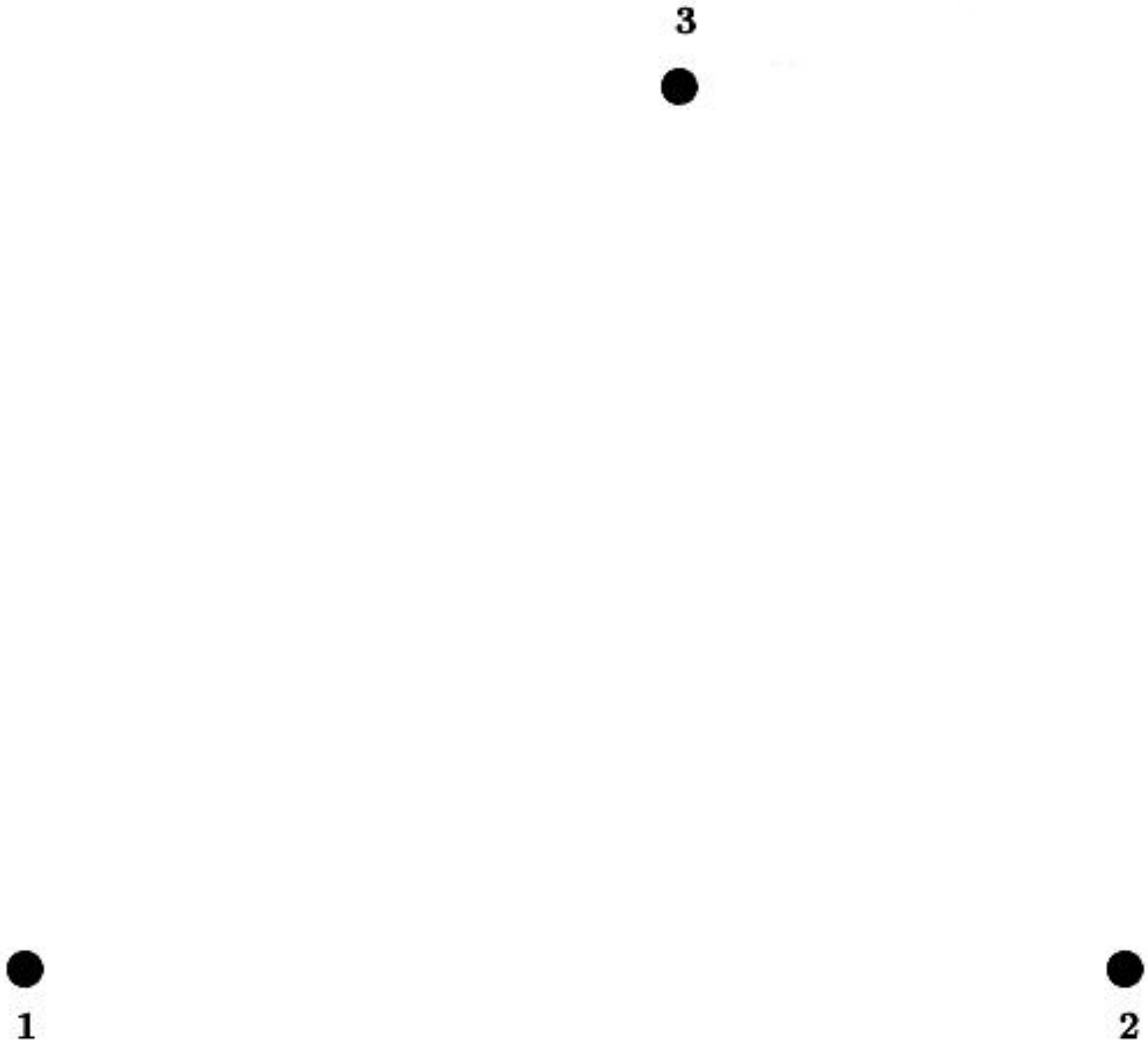
3
1
2

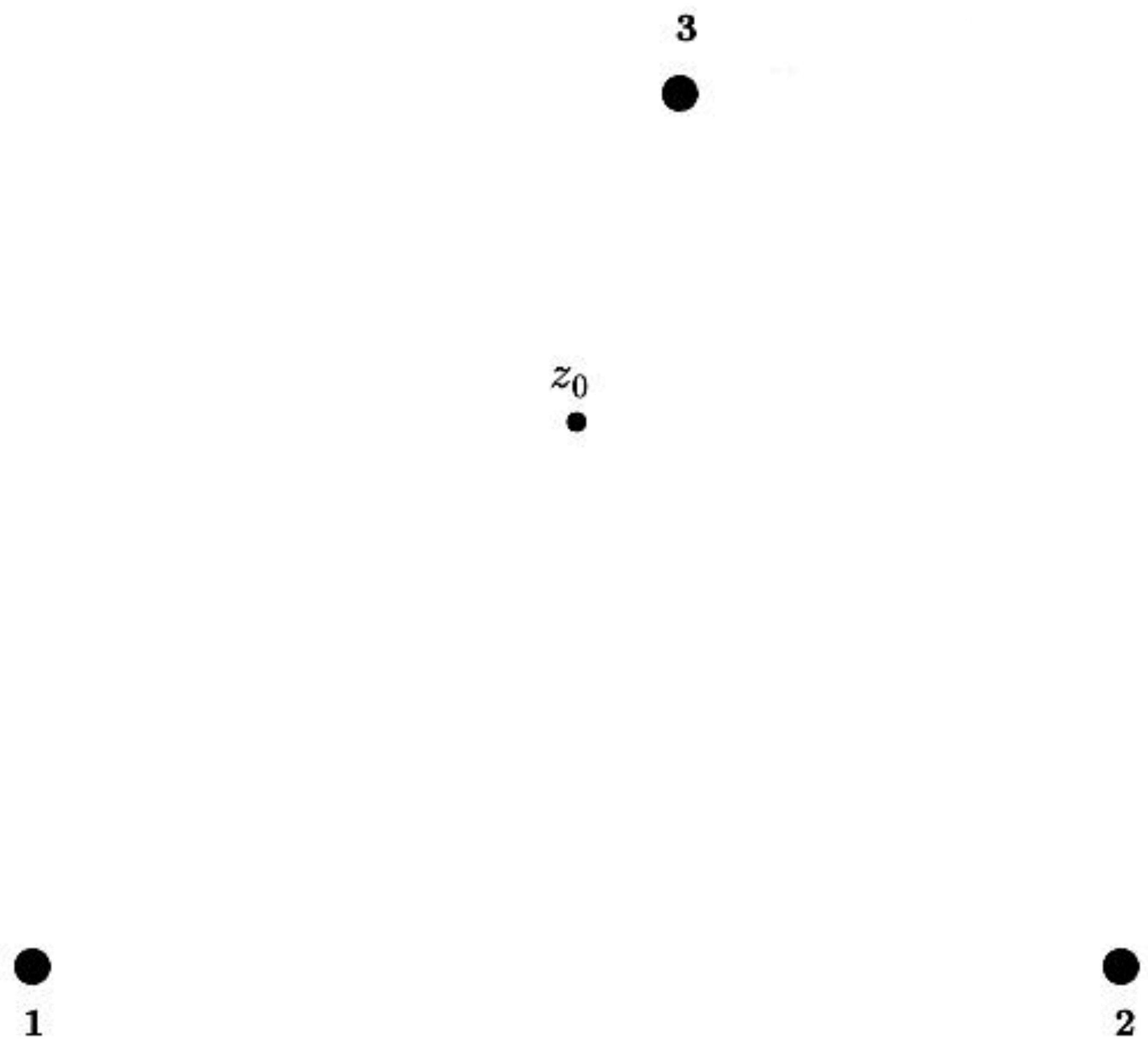


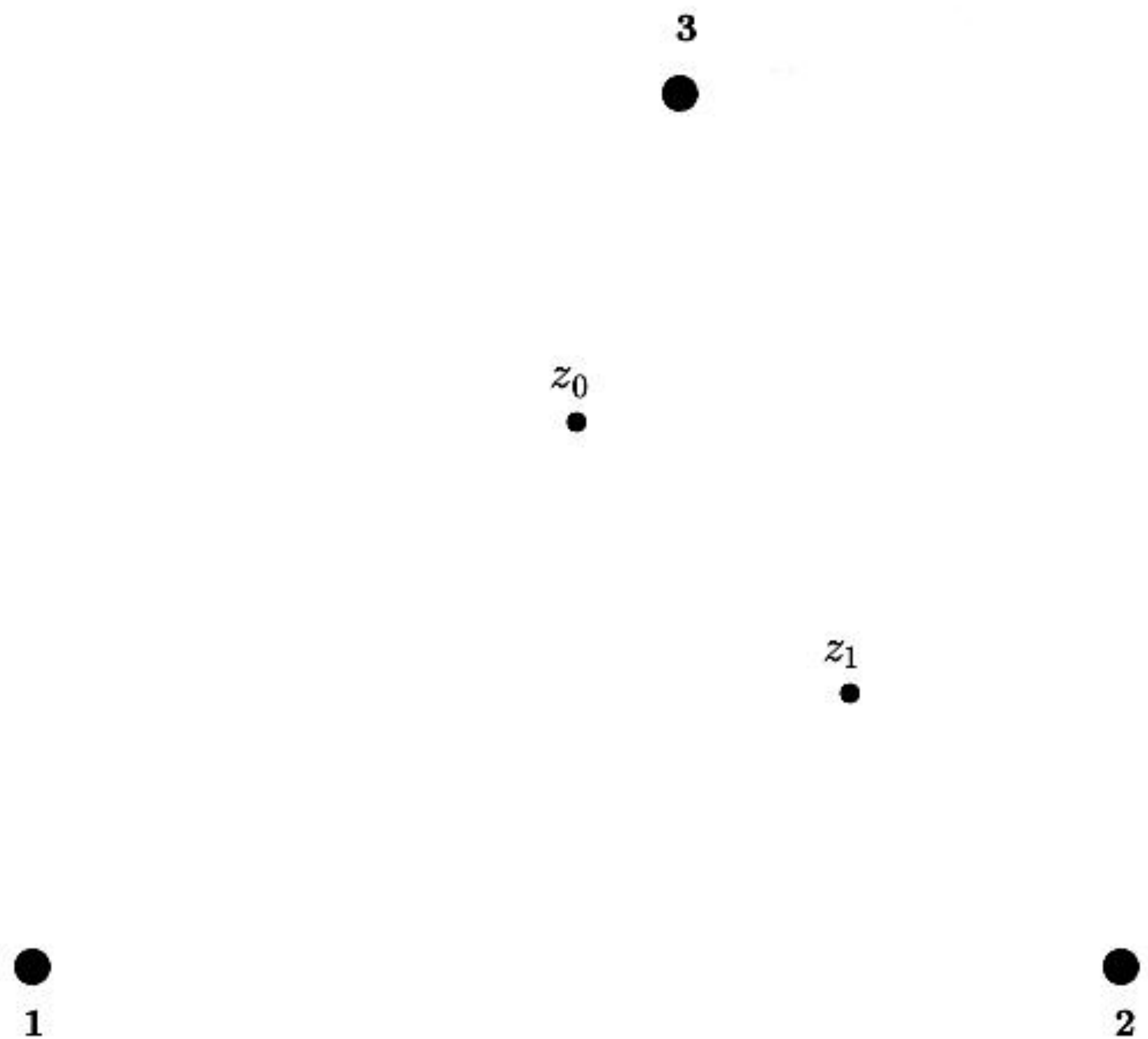
Sierpinski variation

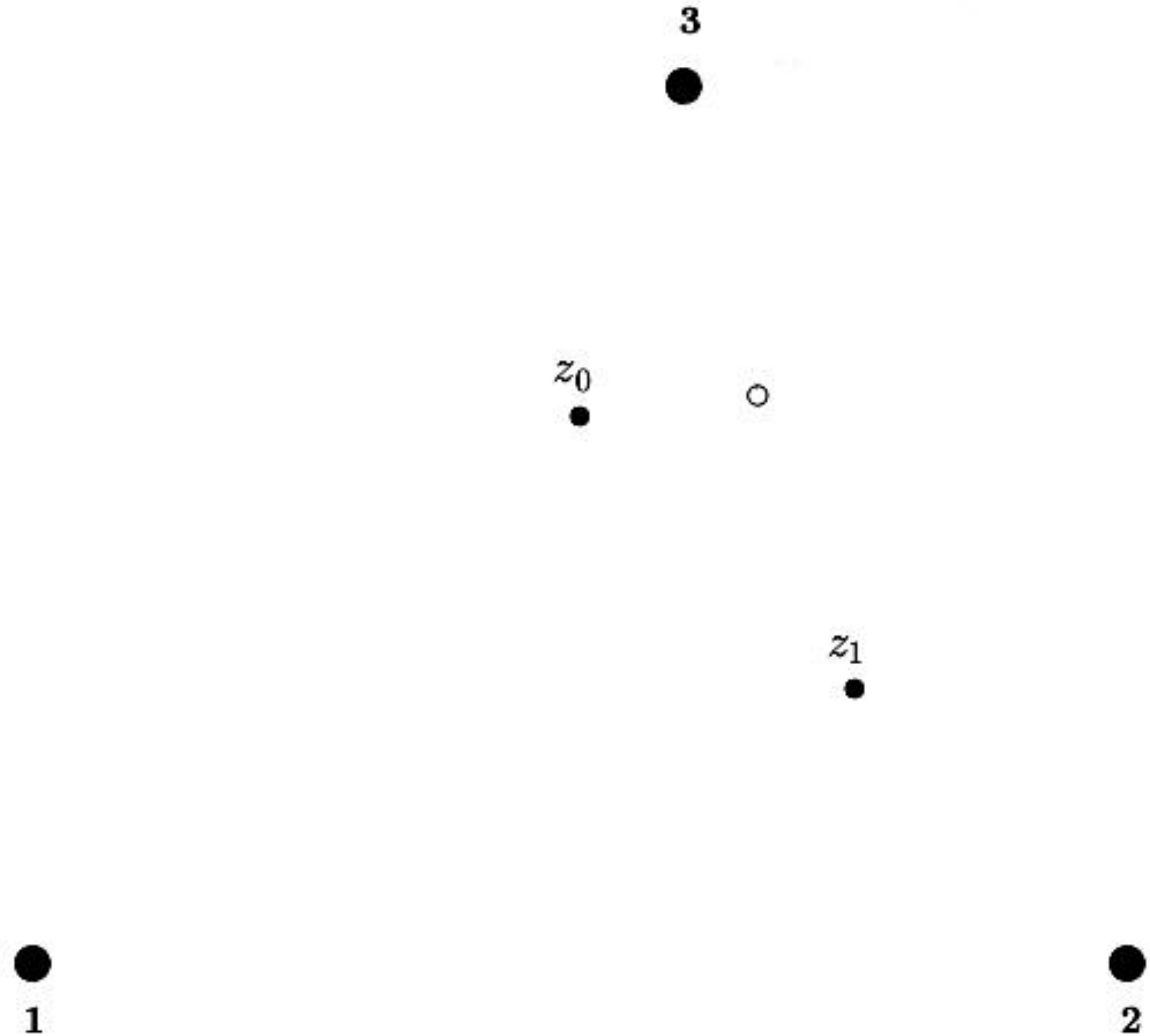
- three pins; 1 and 2 along a horizontal line, 3 above
- choose random number s_i from $\{1, 2, 3\}$
- actions;
  - $s_i =$ 1; move 1/2 distance to pin labelled 1
  - $s_i =$ 2; move 1/2 distance to pin labelled 2
  - $s_i =$ 3; move 1/2 distance to pin 3 and then rotate counterclockwise about pin 3 by 90 degrees

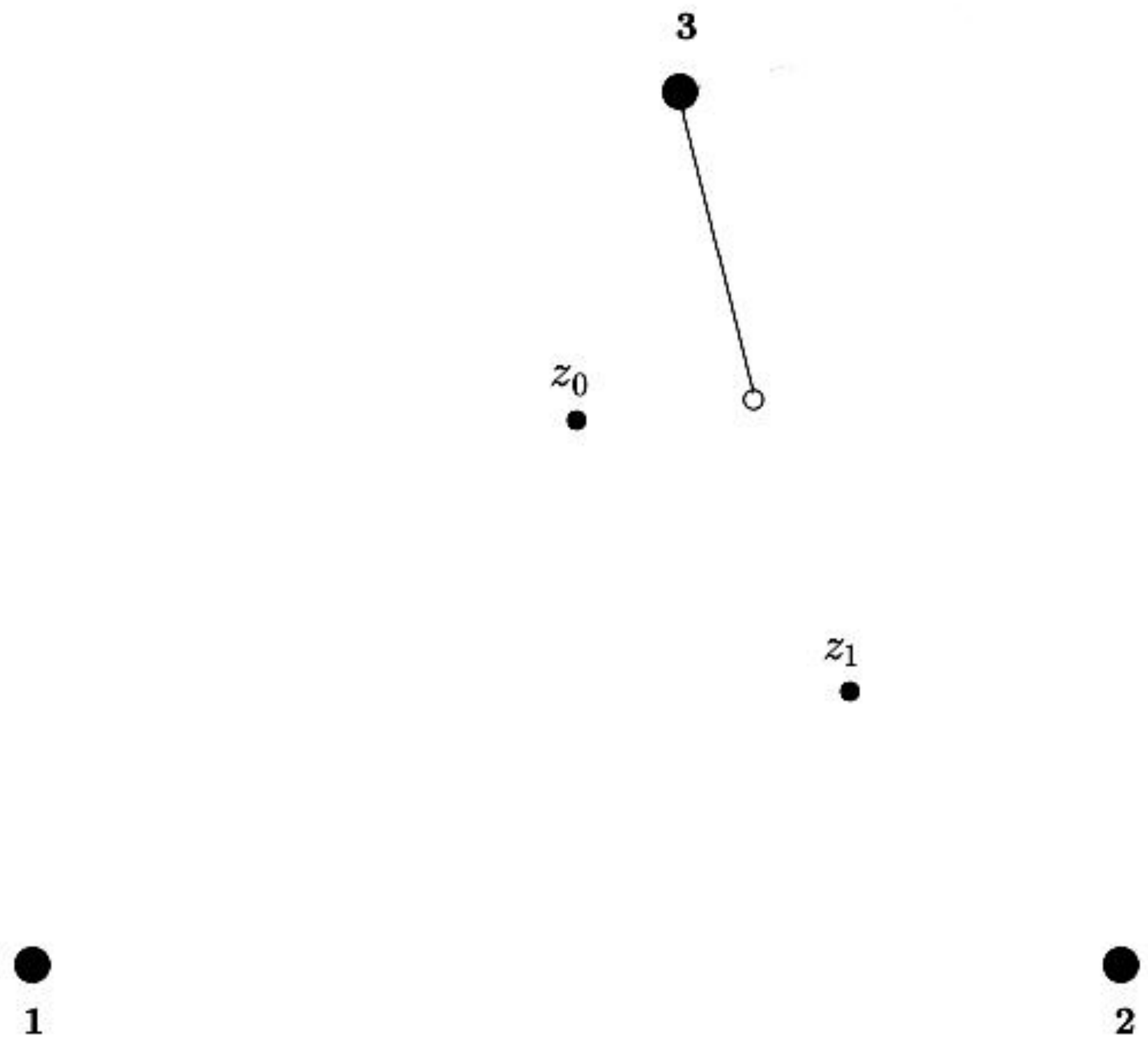



3
1
2

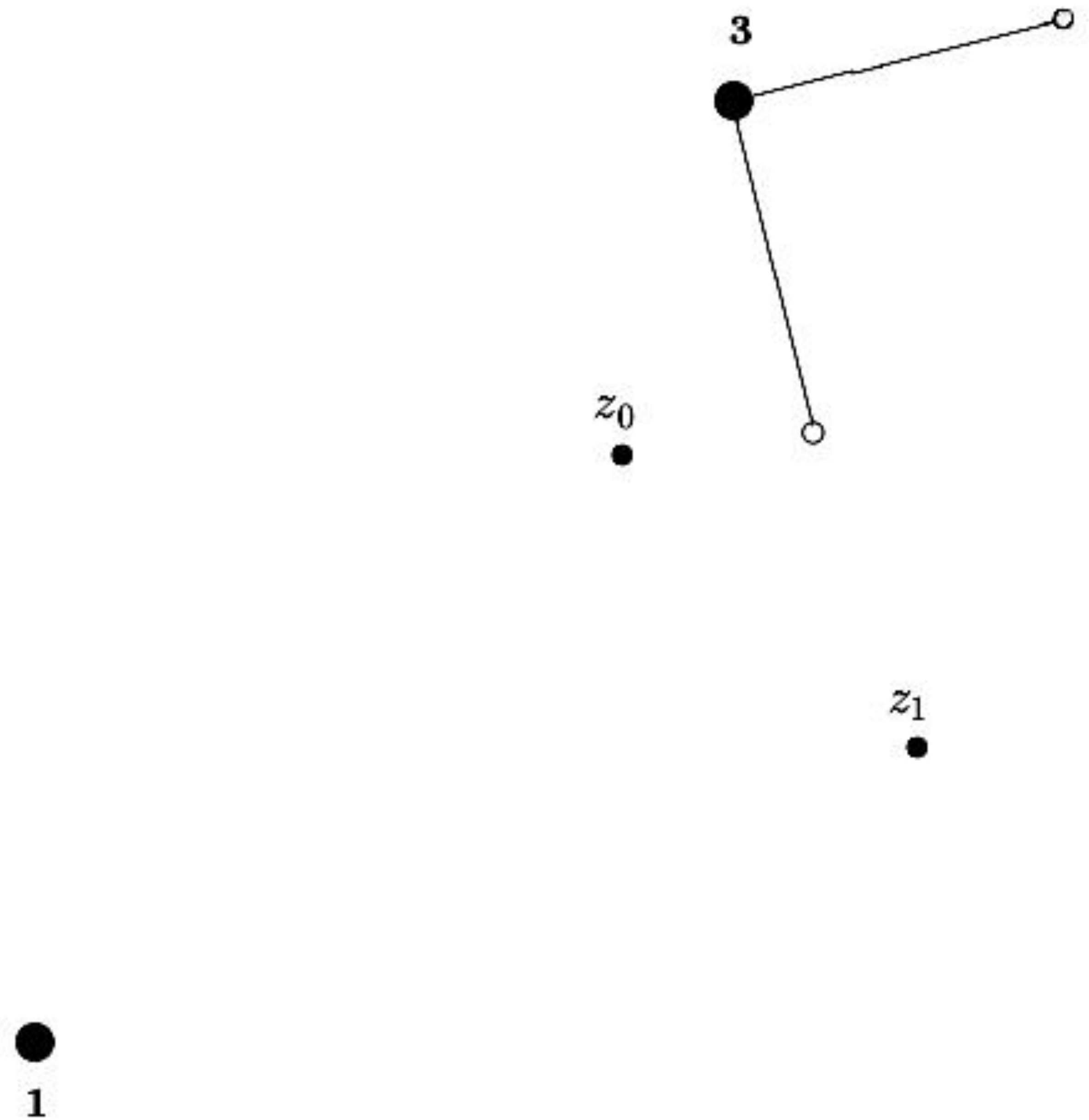

3
z_0
1
2

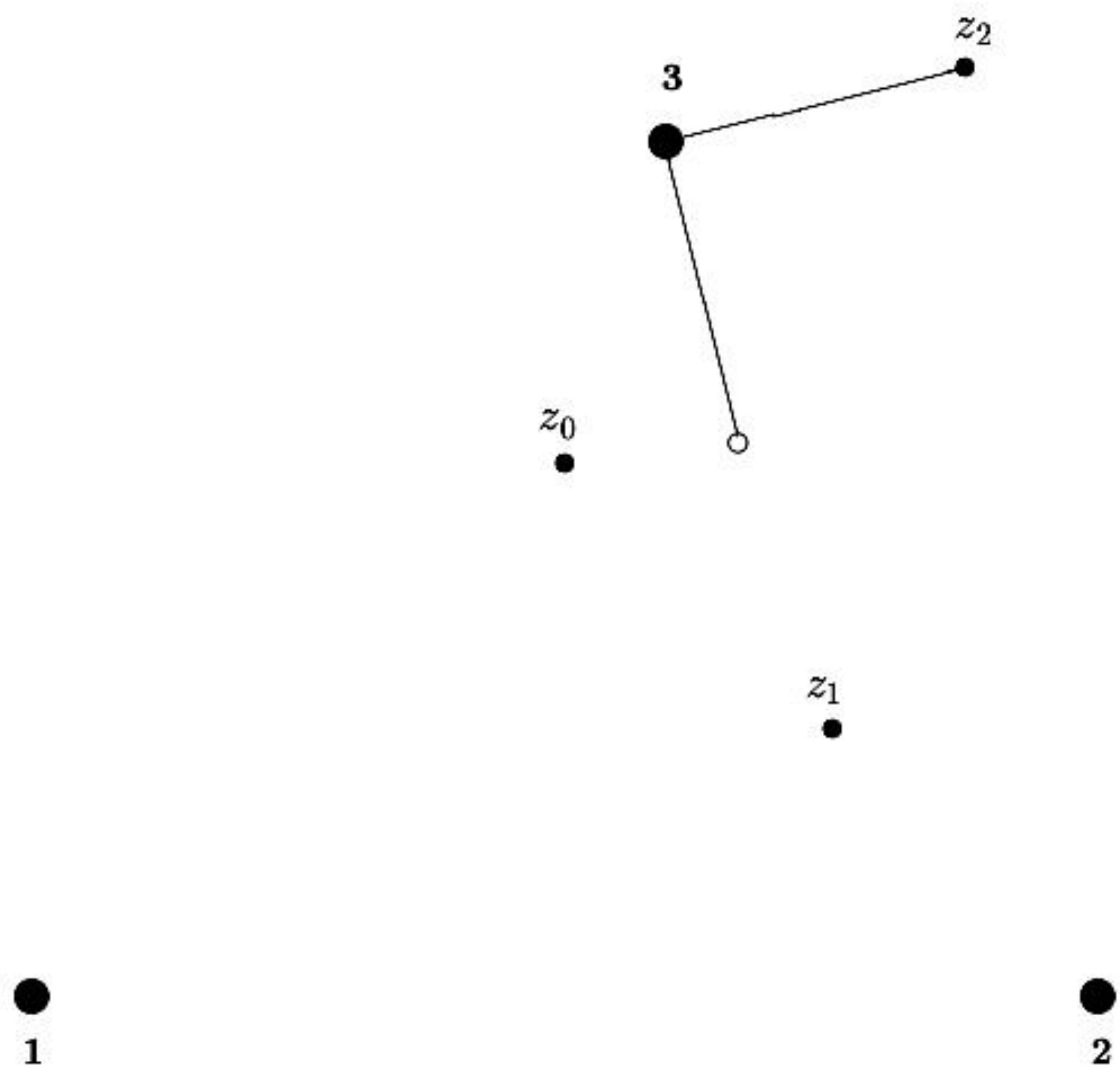

3
z_0
z_1
1
2


3
z_0
z_1
1
2



3
z_0
z_1
1
2


3
z_0
z_1
1
2



z_2
3
z_0
z_1
1
2

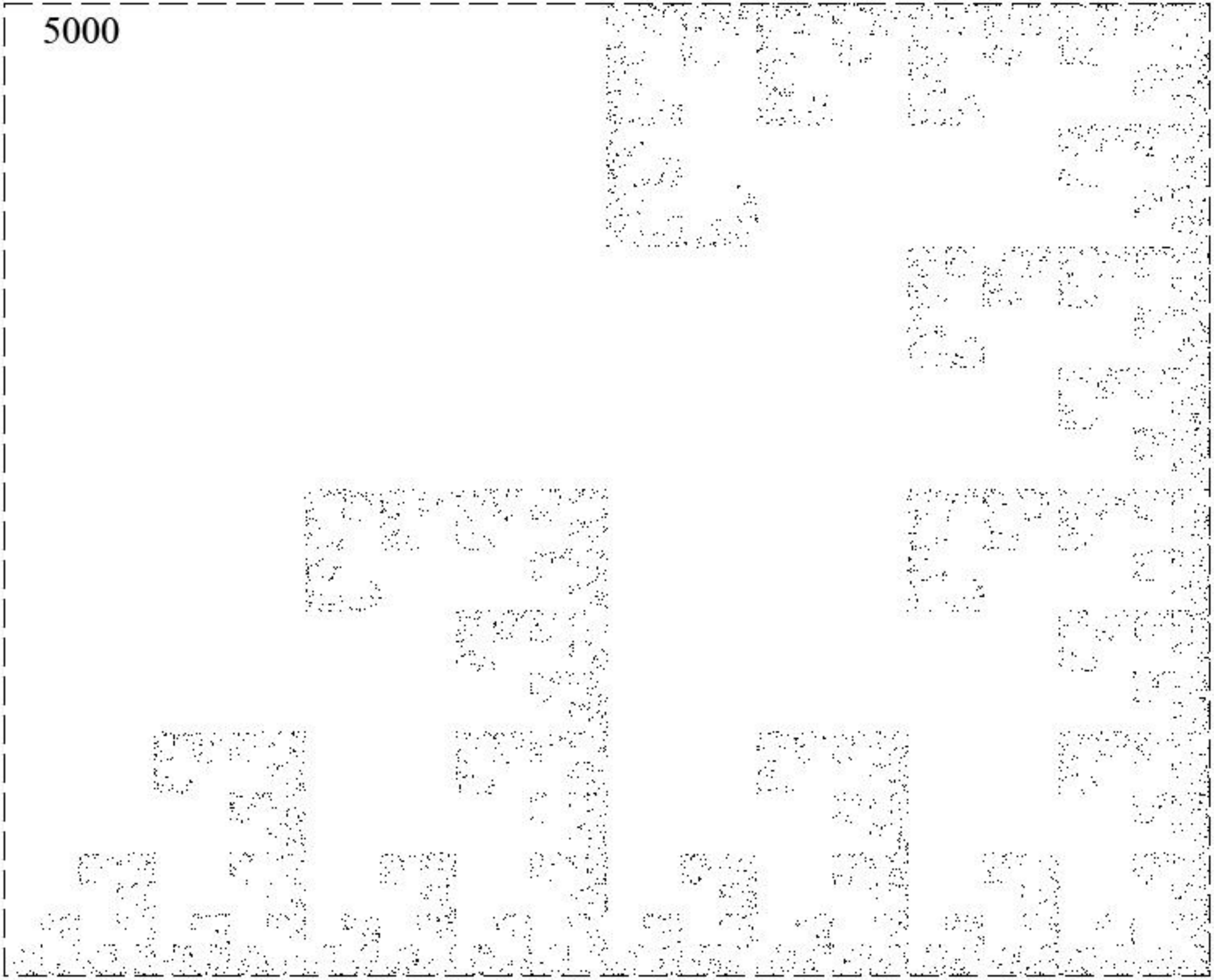
z_2
3
z_0
z_3
z_1
1
2

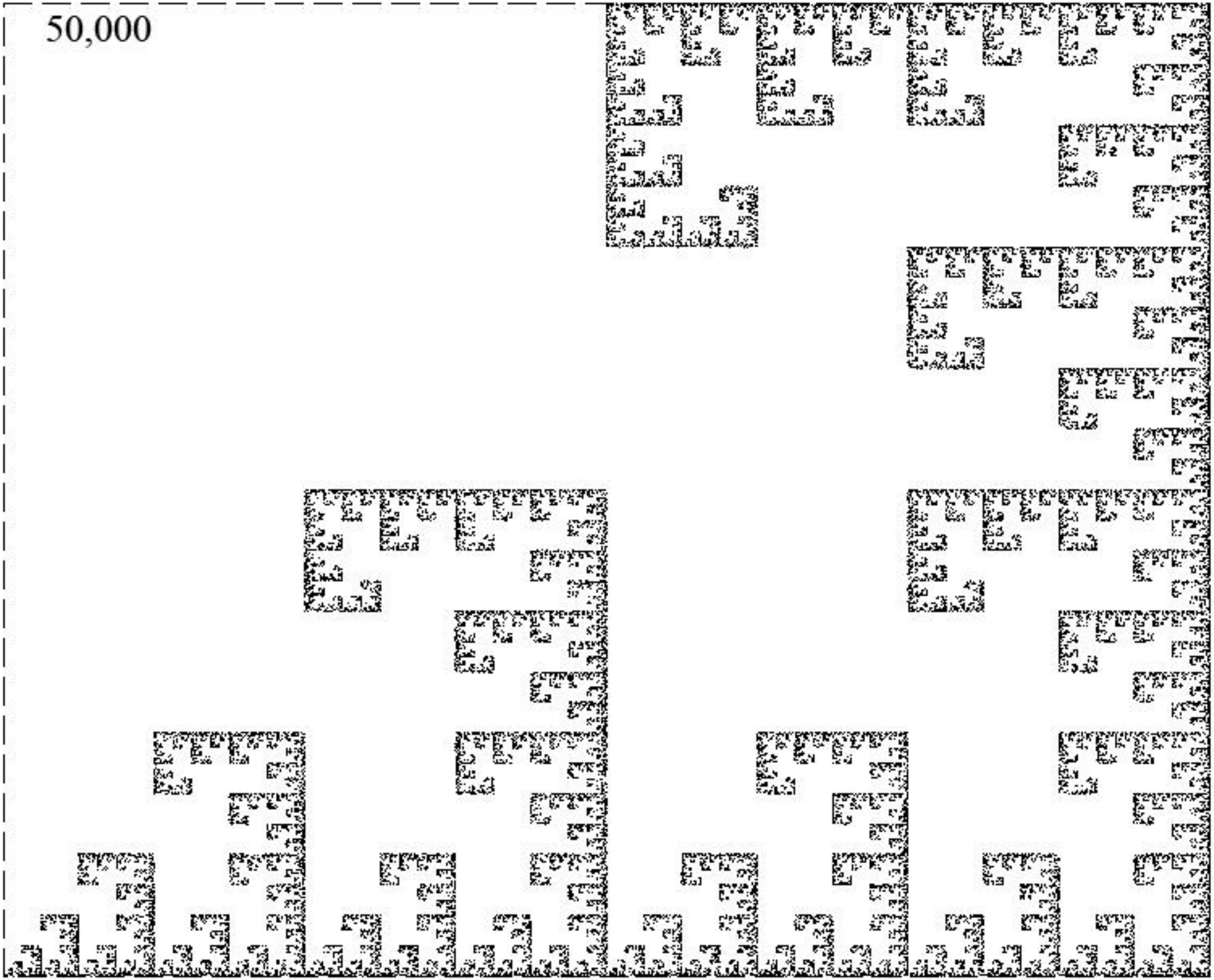
z_2
3
z_0
z_3
z_1
z_4
1
2

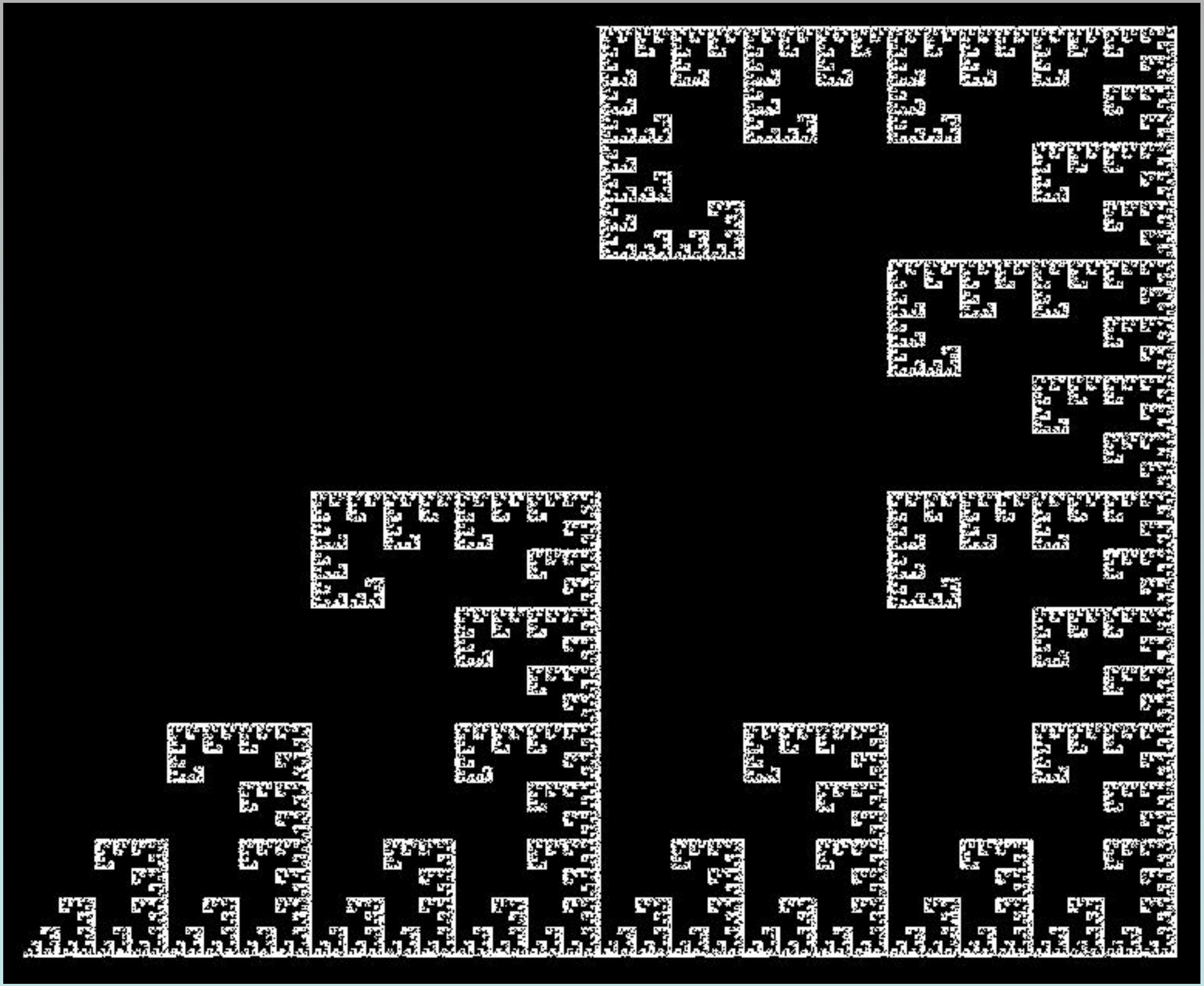

100


500

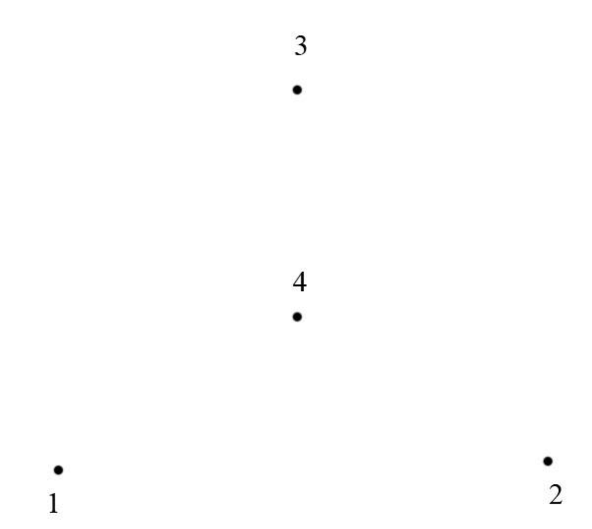

1000


5000


50,000

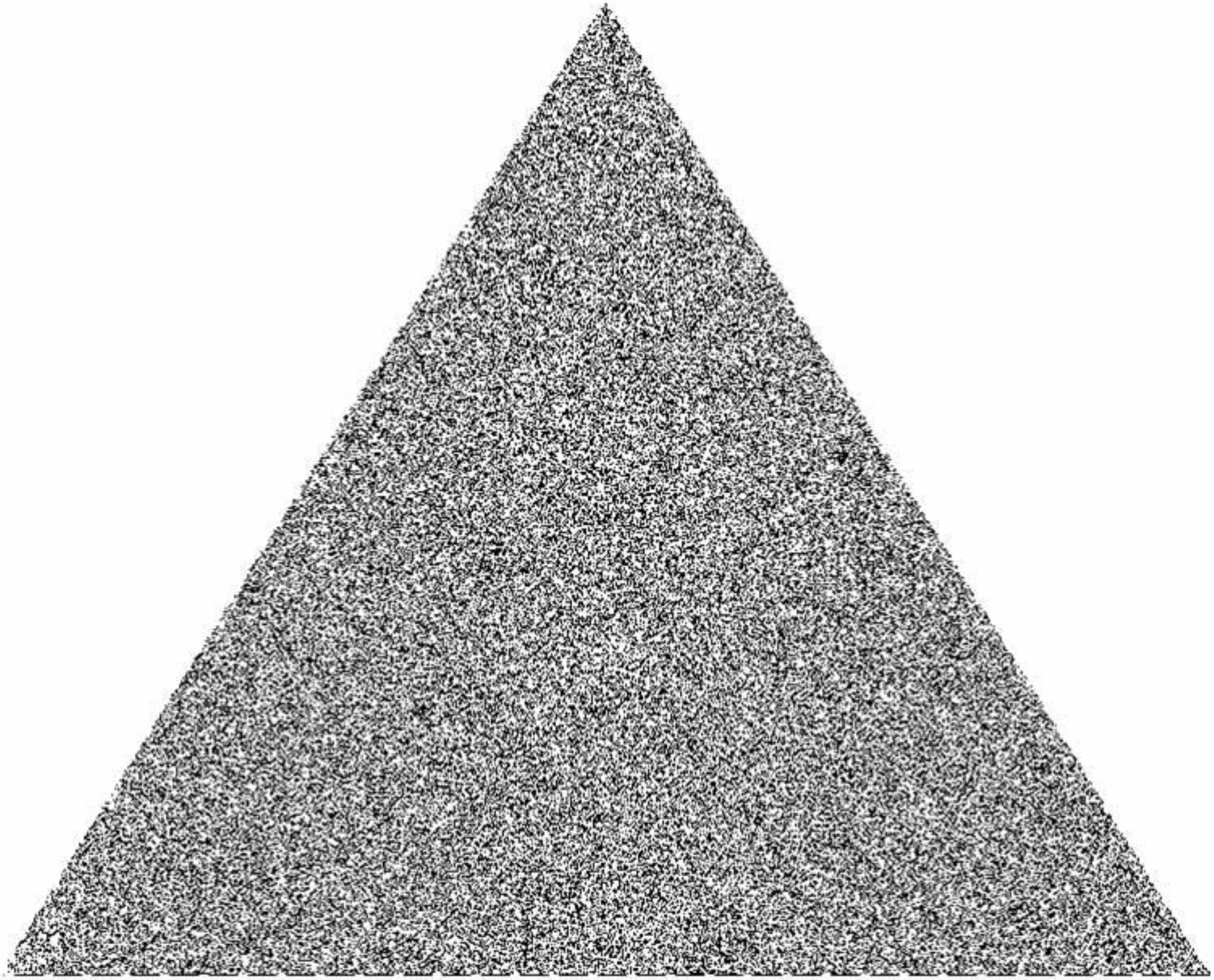


Another chaos game

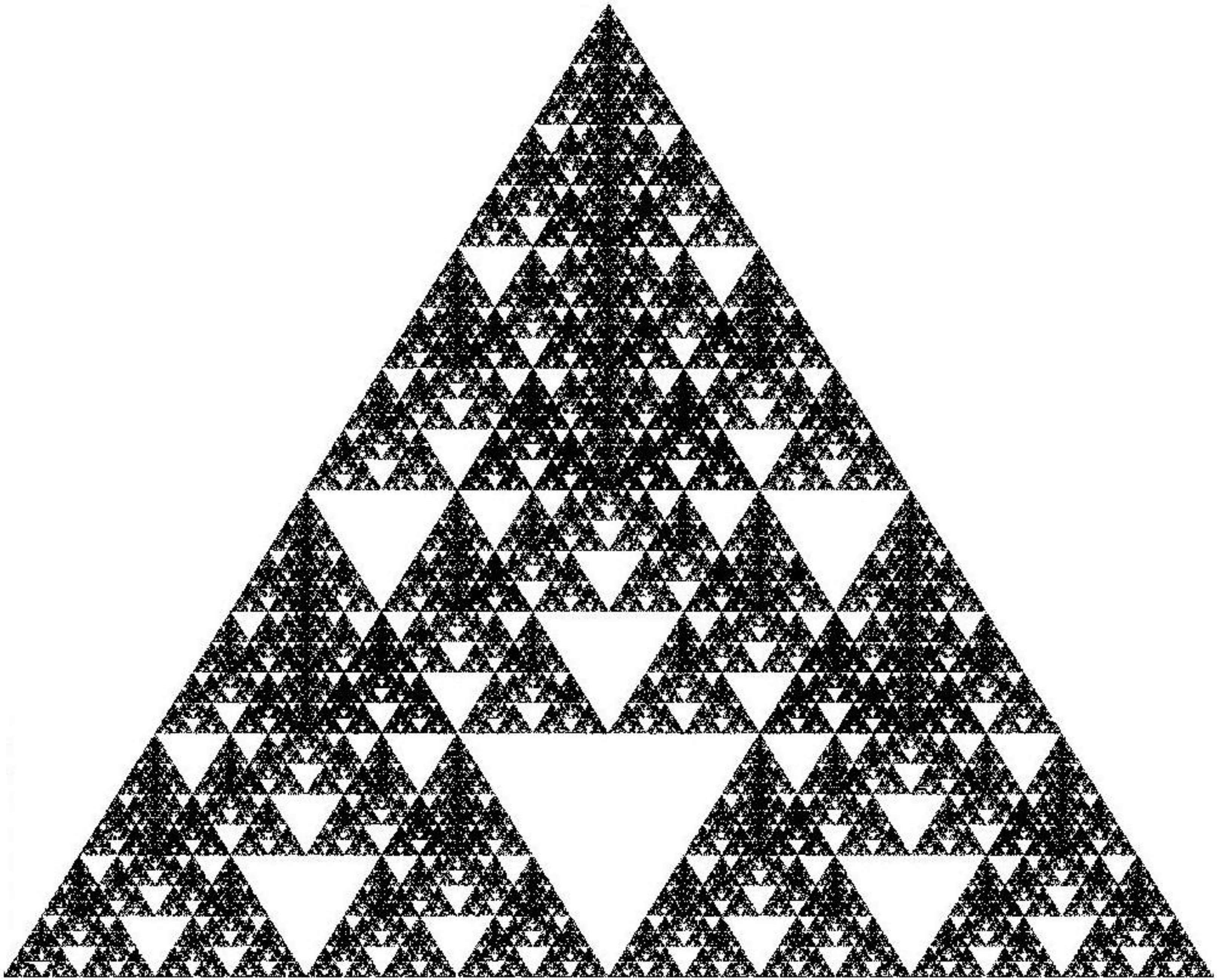


Full Triangle

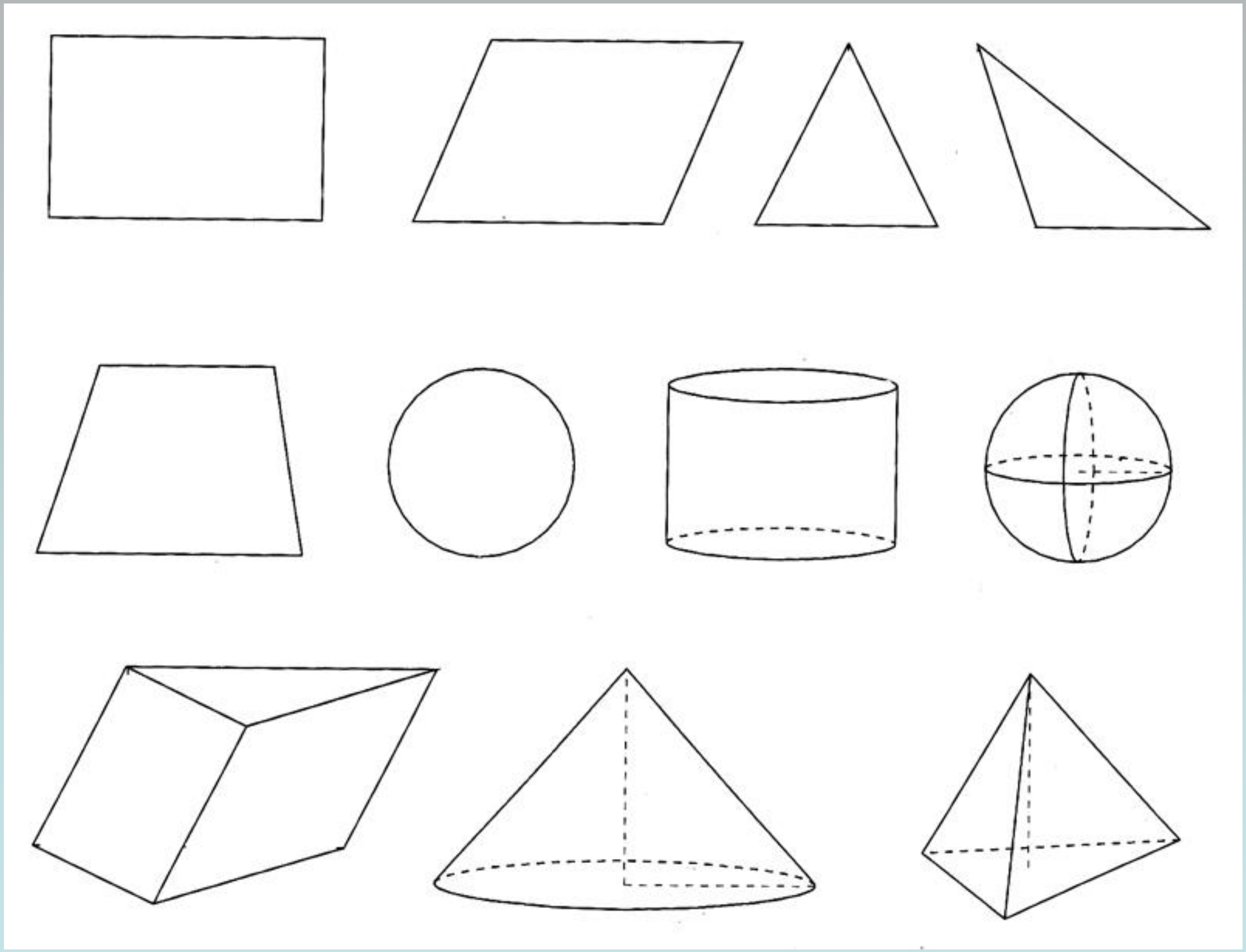
- three pins 1, 2, 3, arranged at vertices of equilateral triangle, pin 4 at the centre
- choose random number s_i from $\{1, 2, 3, 4\}$
- actions;
  - $s_i = 1, 2, 3$; move 1/2 distance from current game point to pin s_i
  - $s_i = 4$; move 1/2 distance from current game point to pin 4 and then rotate 180 degrees about pin 4



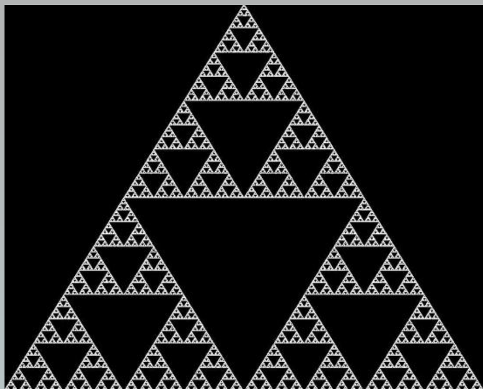
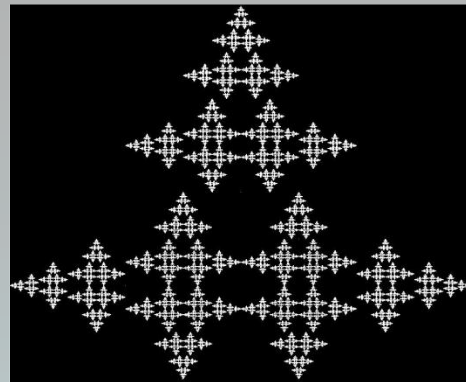
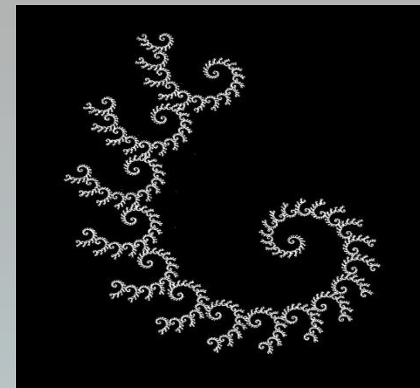
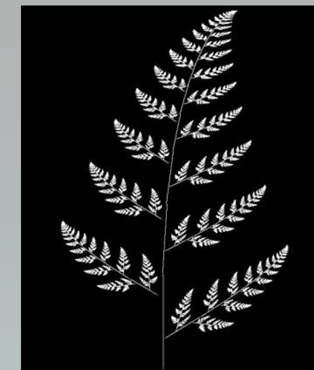
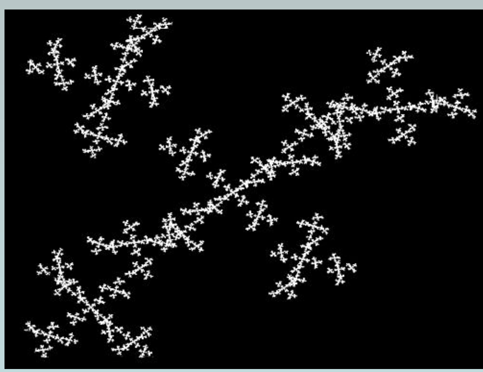
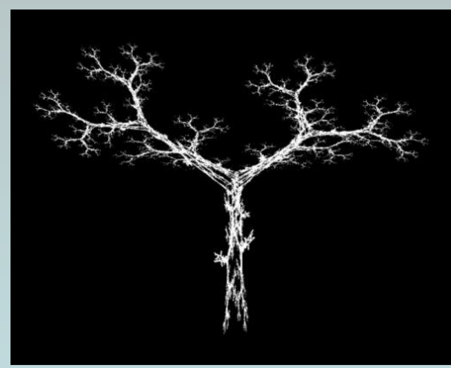
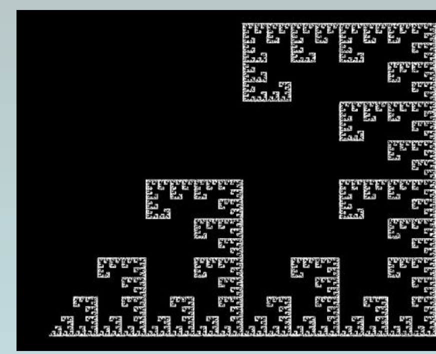
What if action 4 was; “move ½ distance towards pin 4” only (i.e., no rotation)……

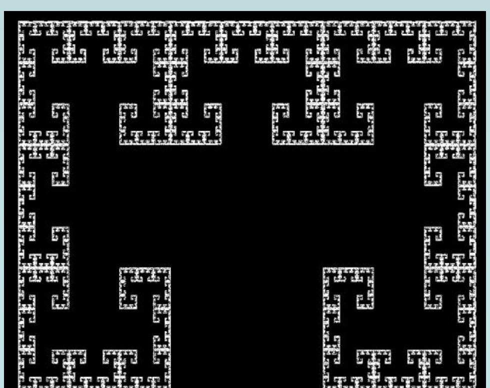
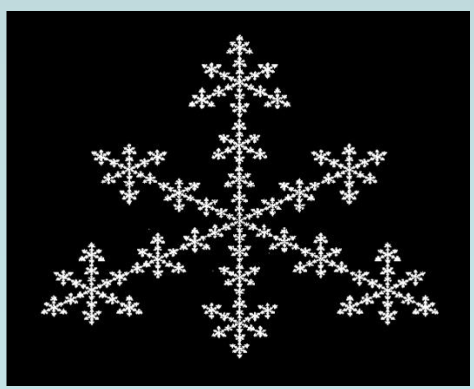
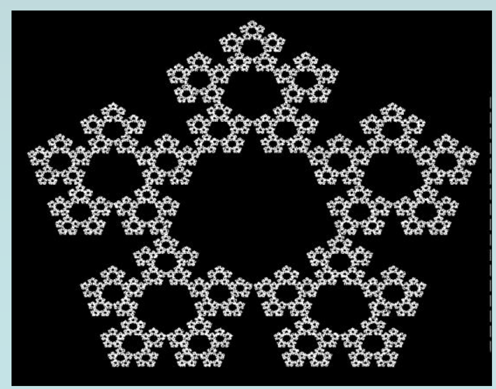
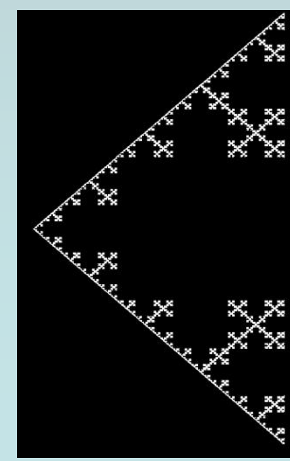


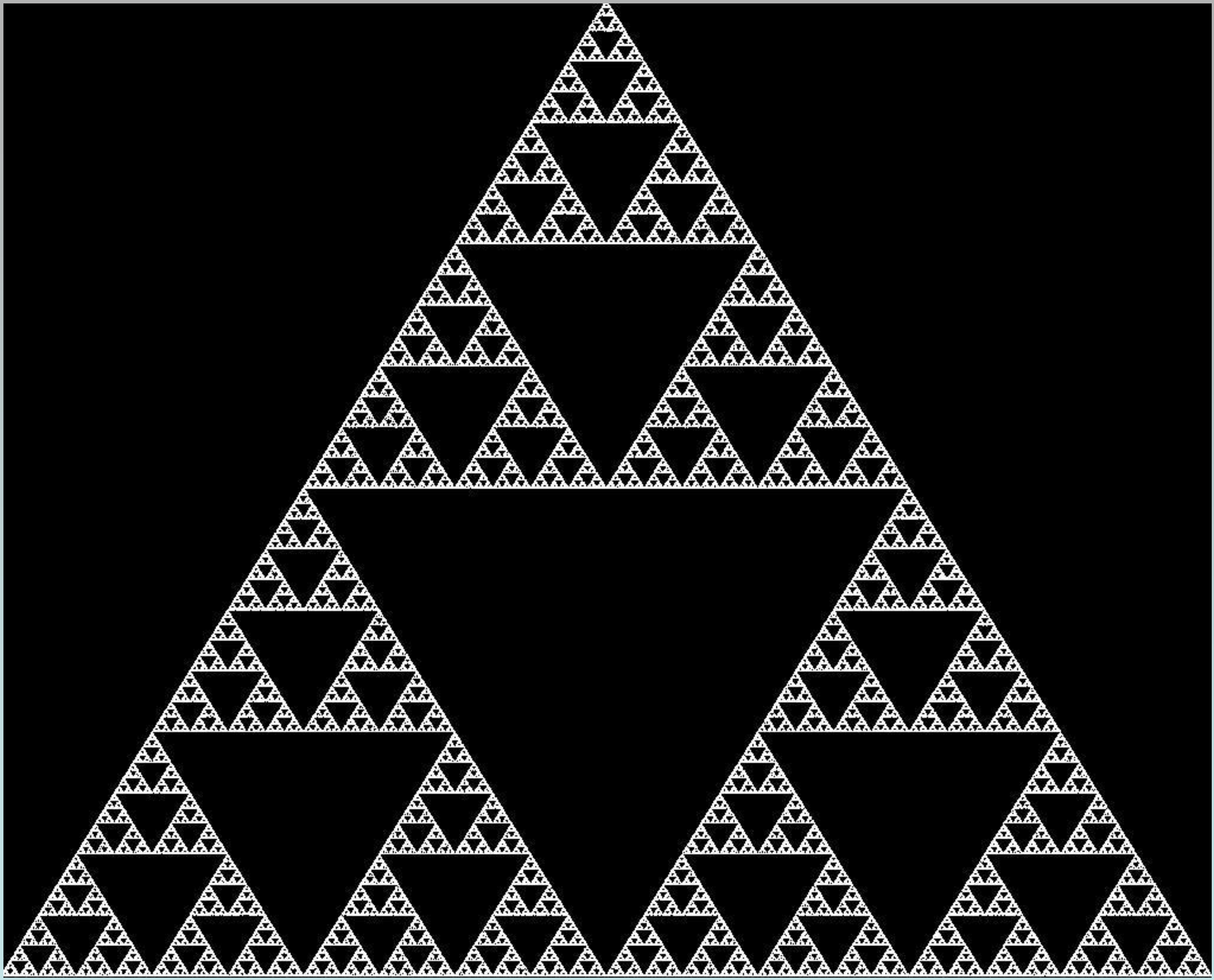
Regular (Euclidean) geometry

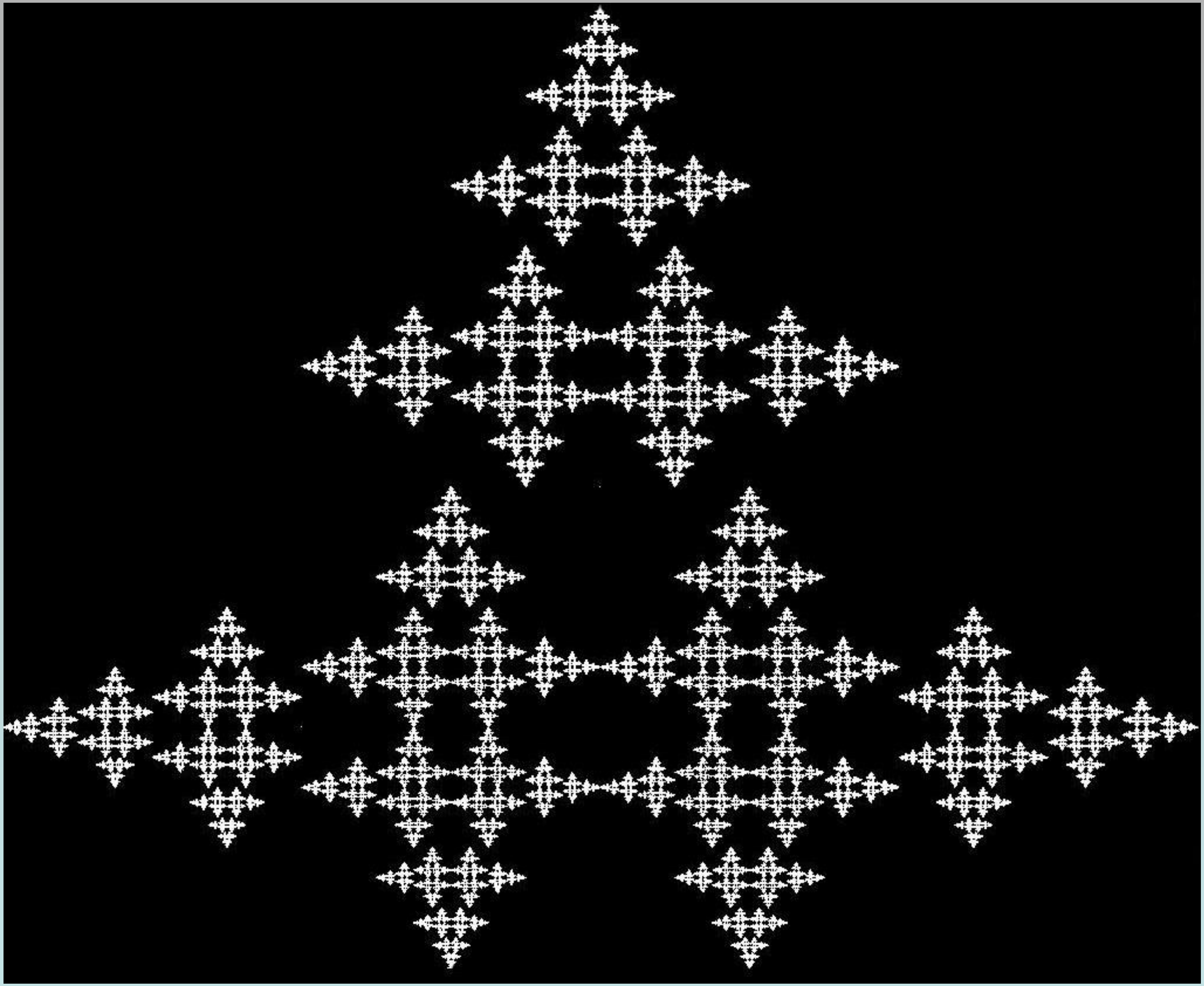


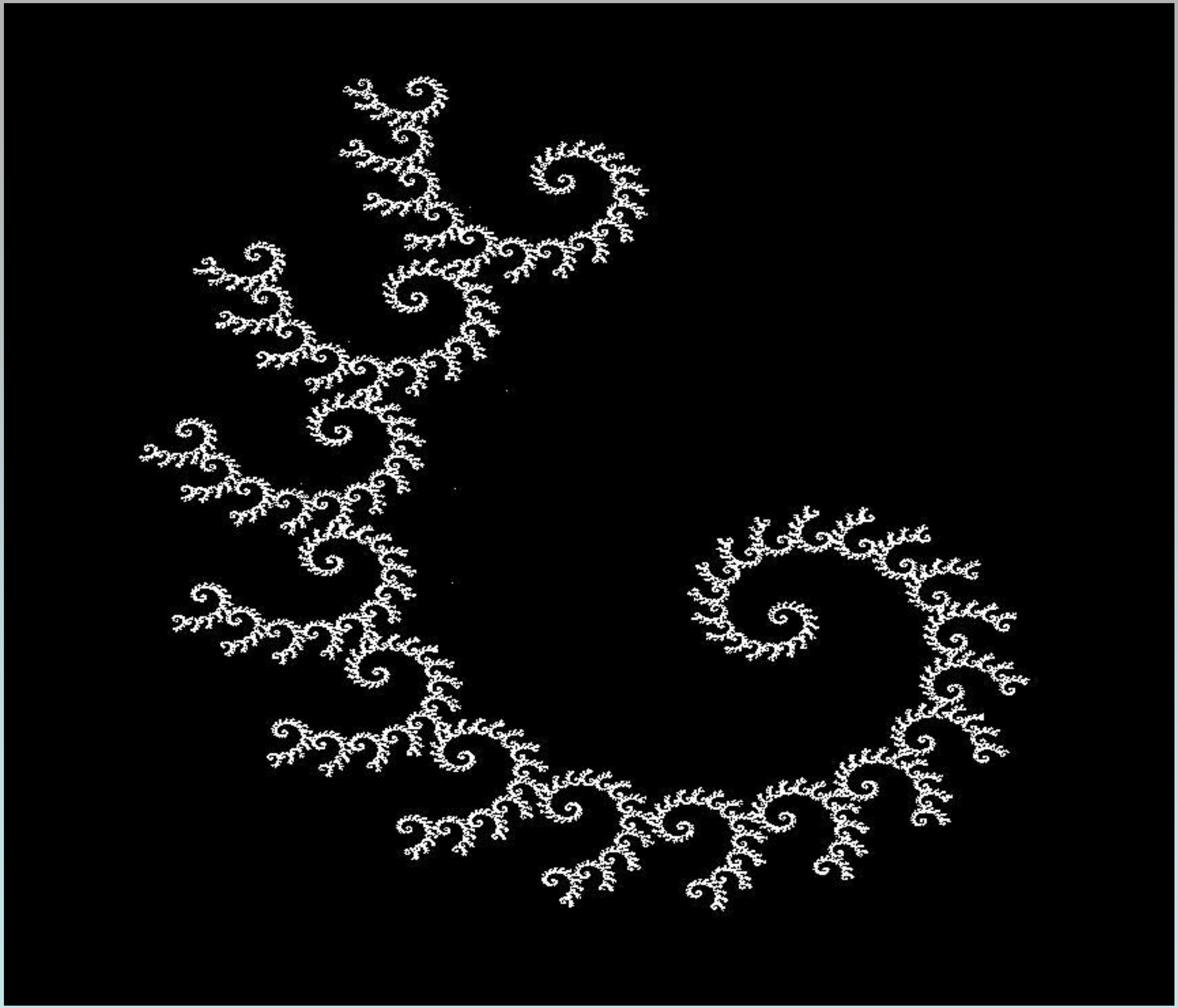
Fractal geometry. . . .

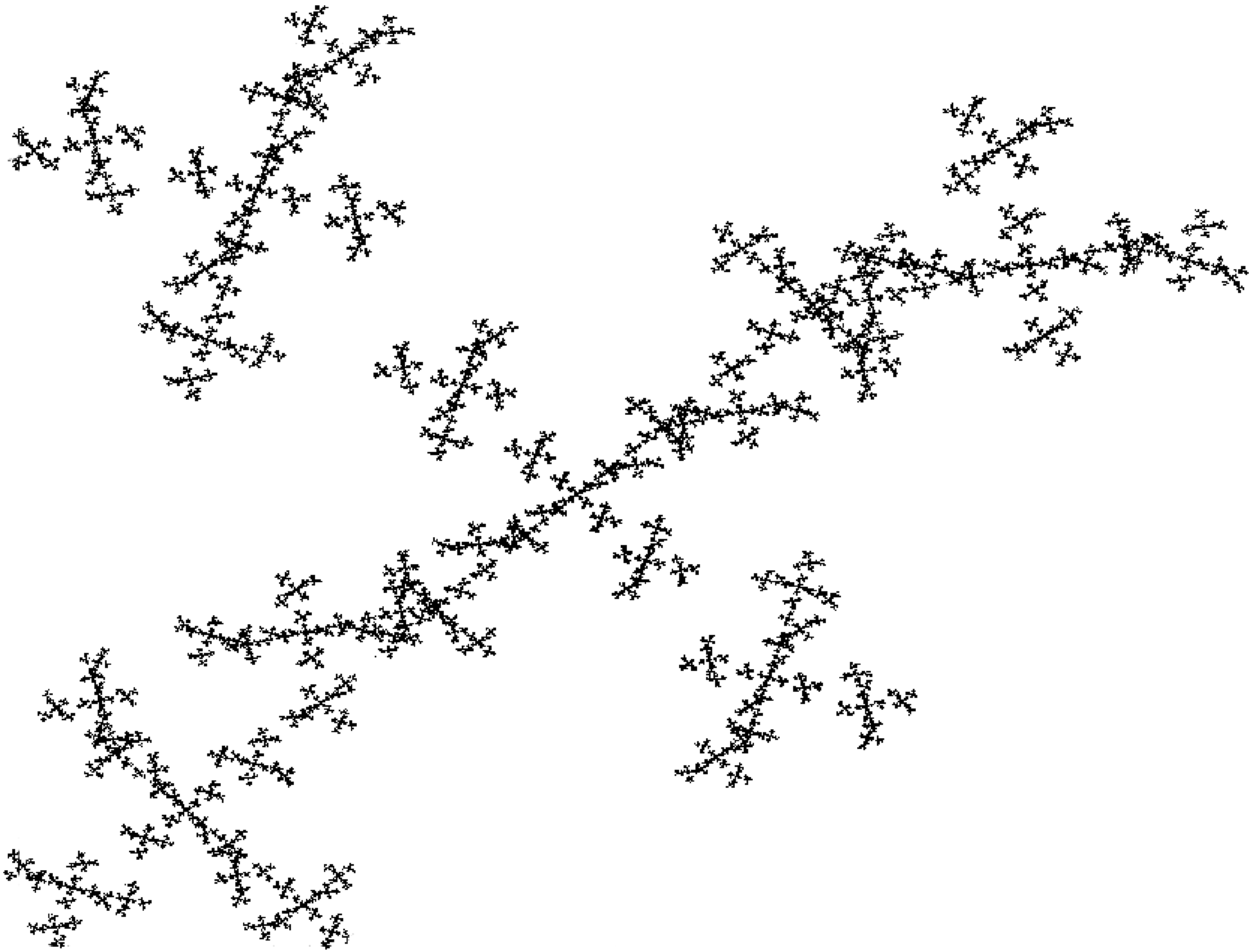








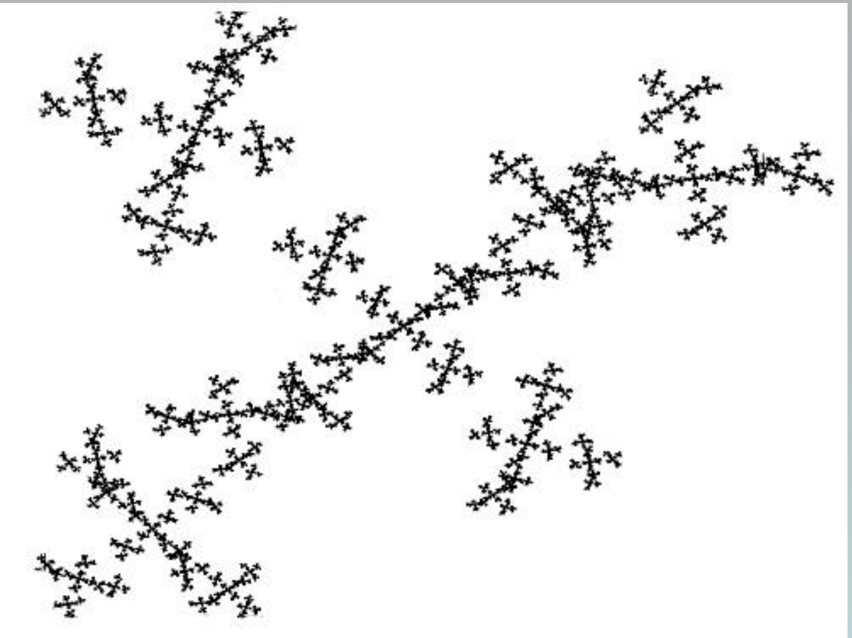






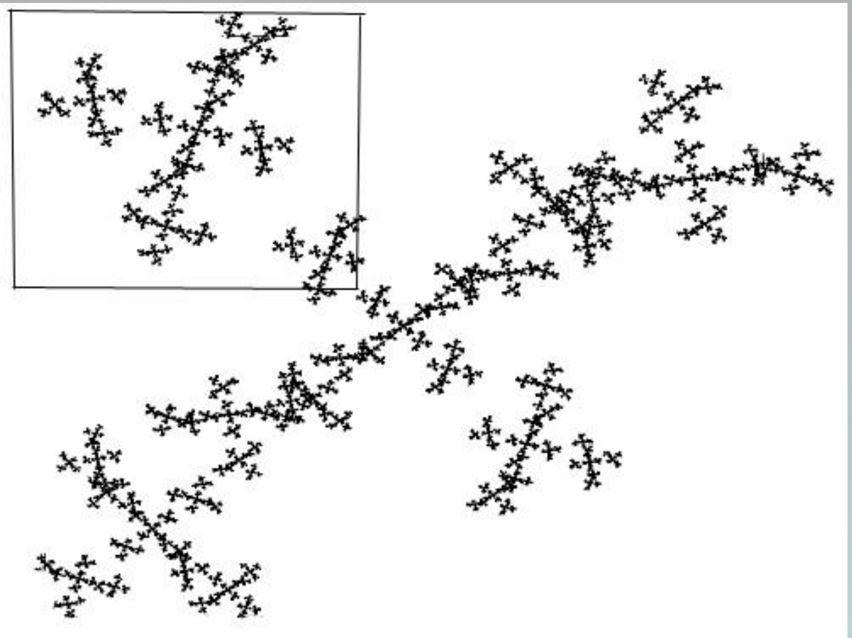


Self-Similarity



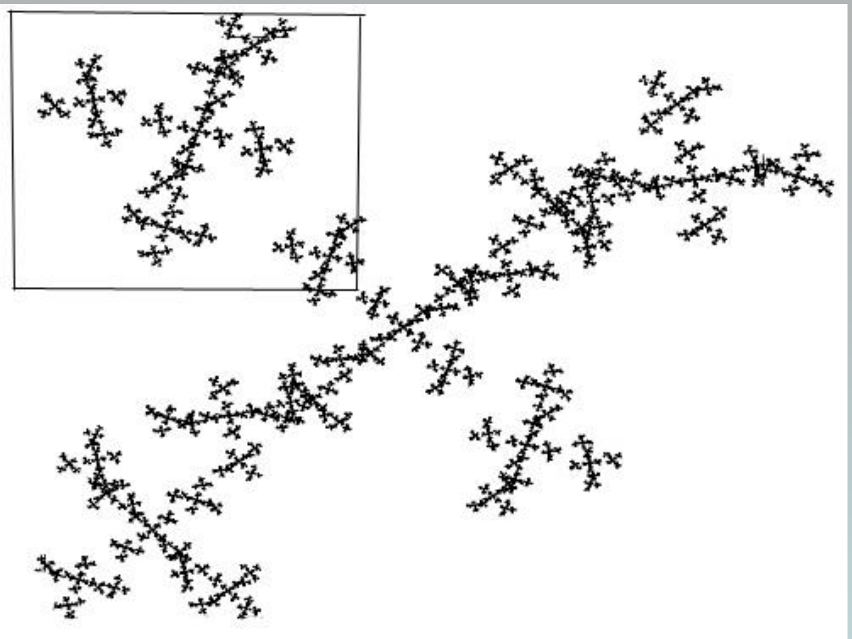
The whole fractal

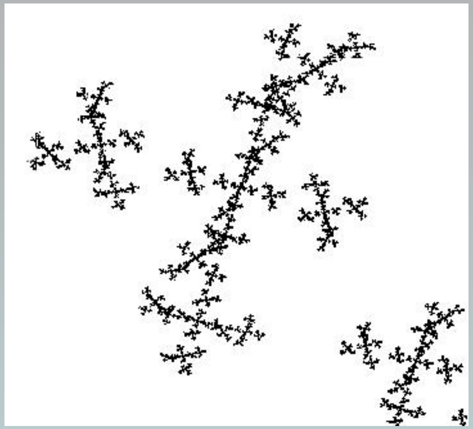
Self-Similarity



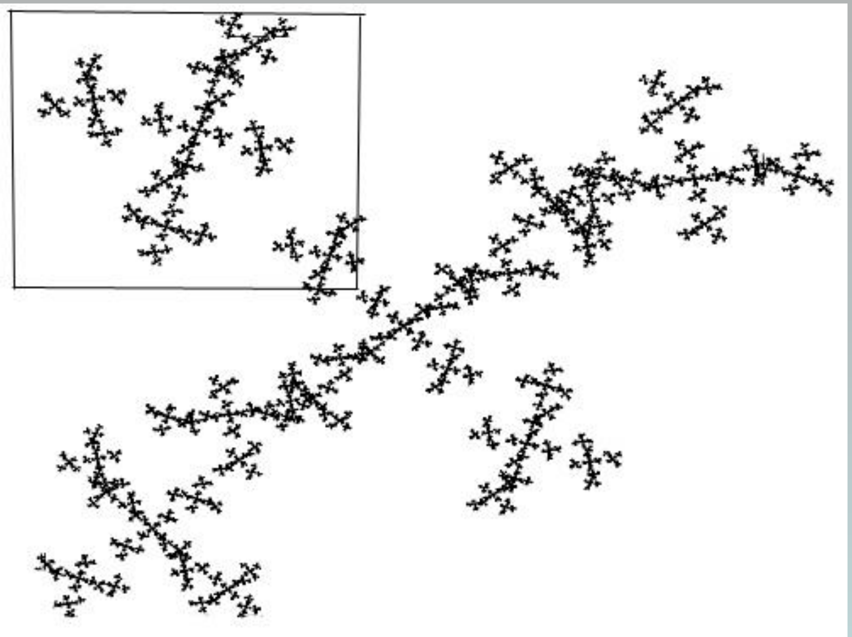
Pick a small copy of it

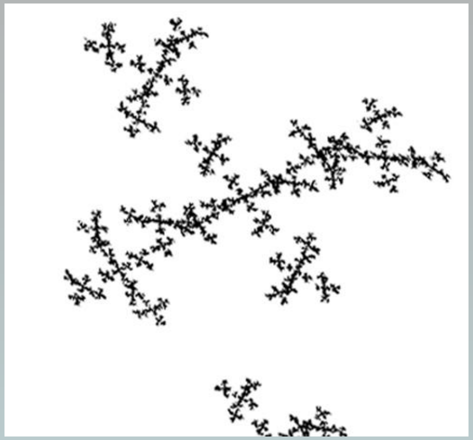
Self-Similarity





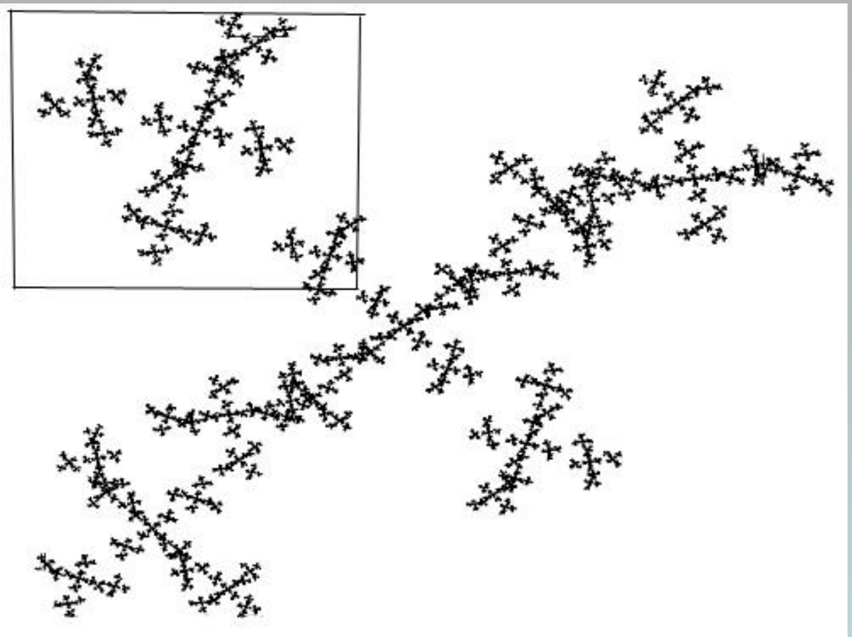
Self-Similarity

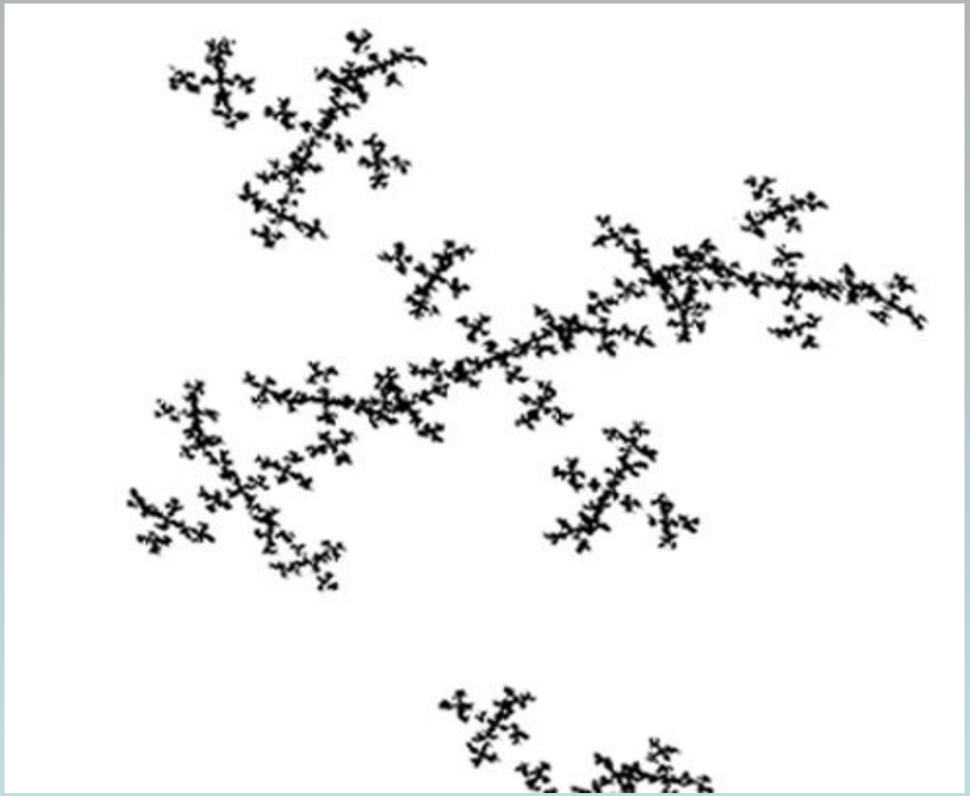




Rotate it

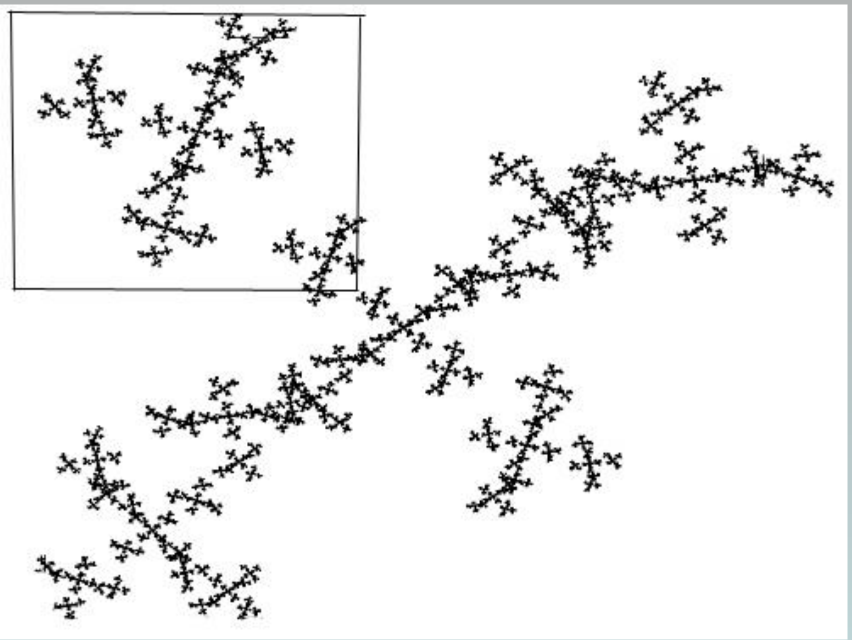
Self-Similarity

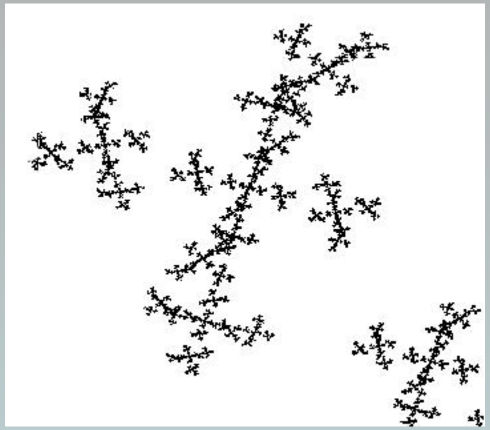


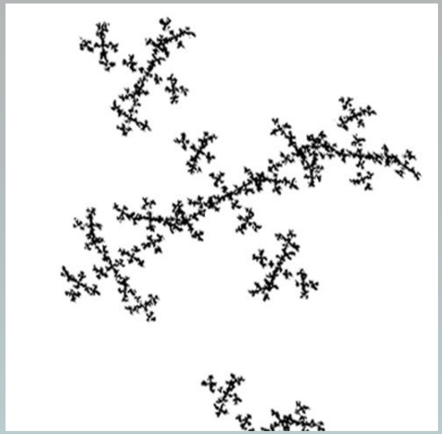


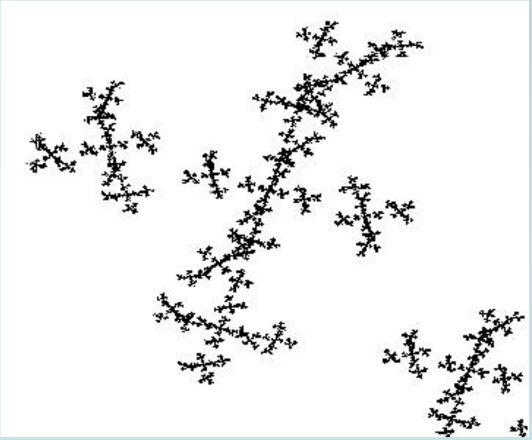
Blow it up: You get the same thing

Self-Similarity



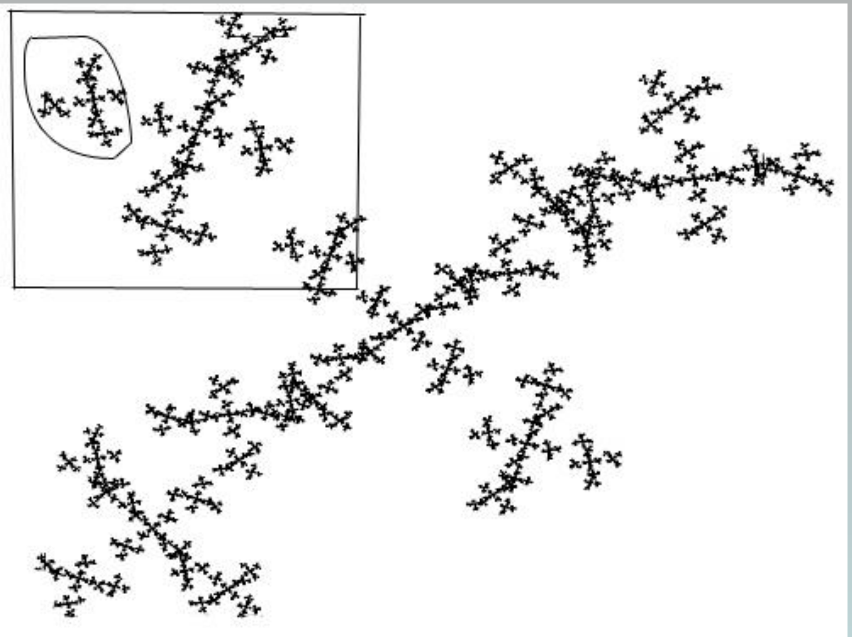


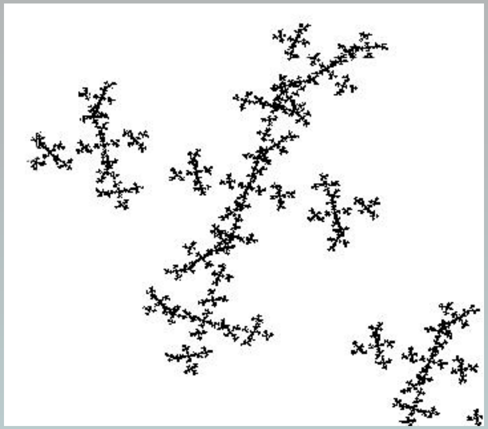


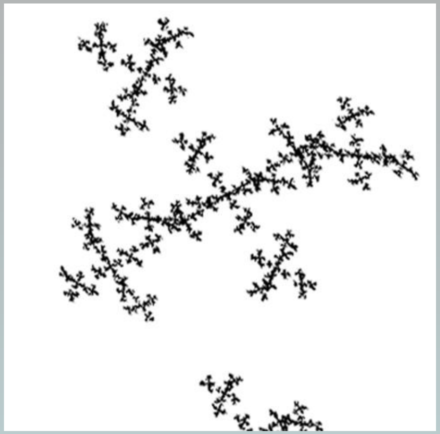


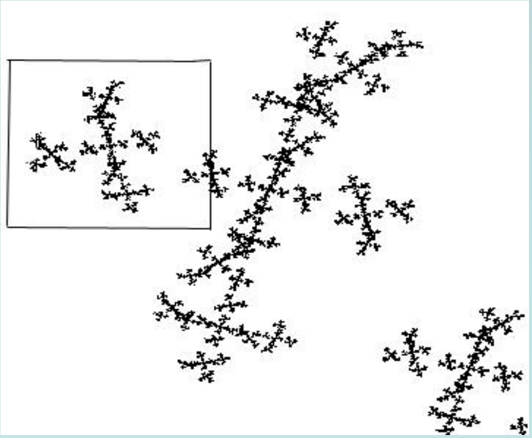
Now continue with that piece

Self-Similarity



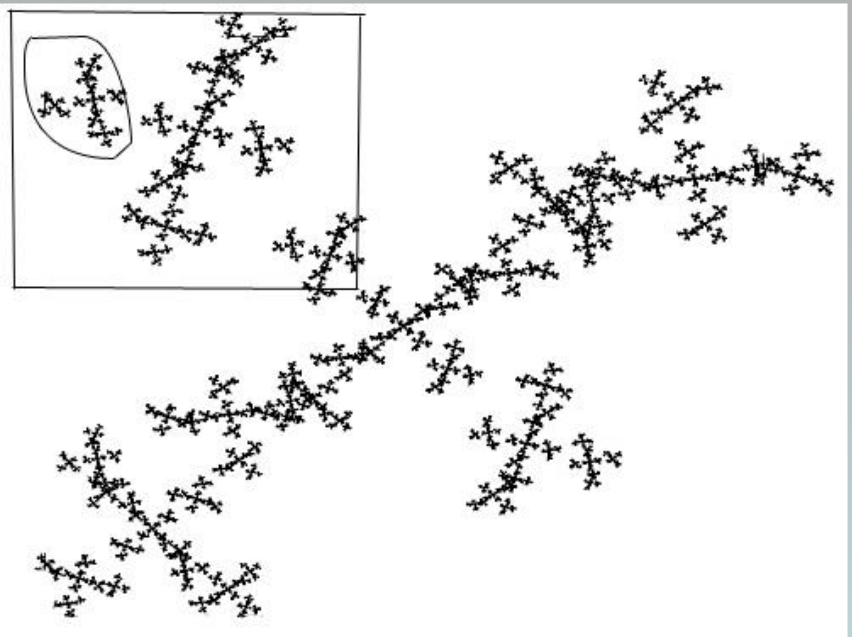


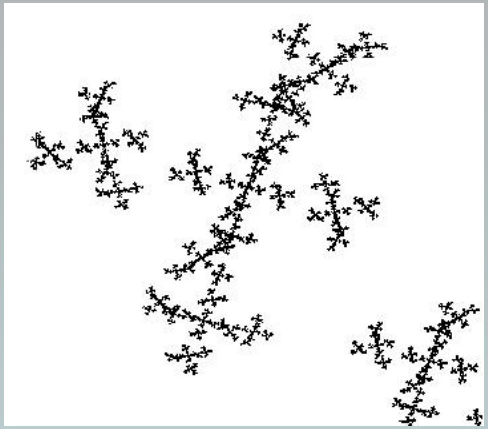


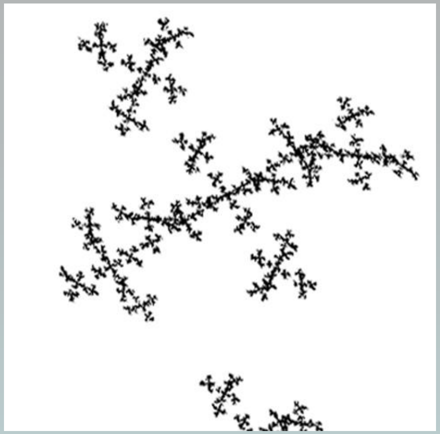


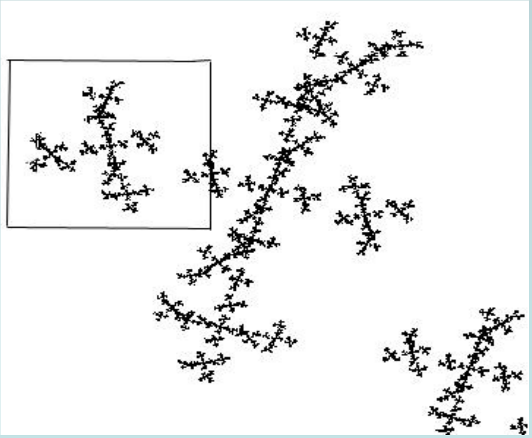
You can find the same small piece on this small piece…

Self-Similarity

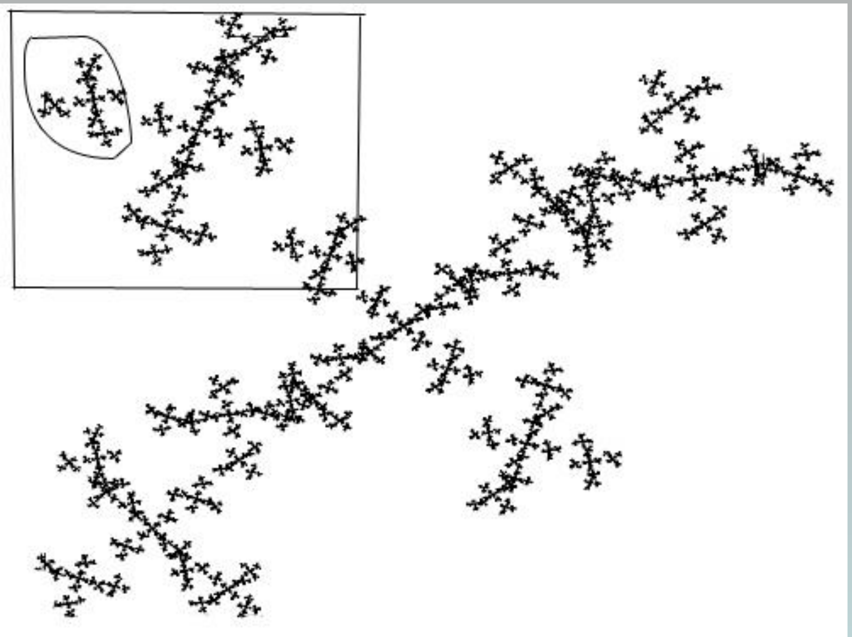


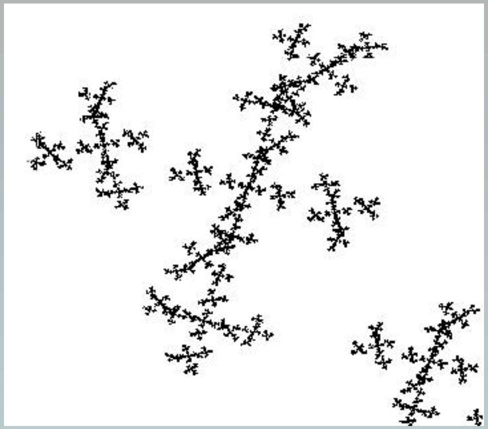


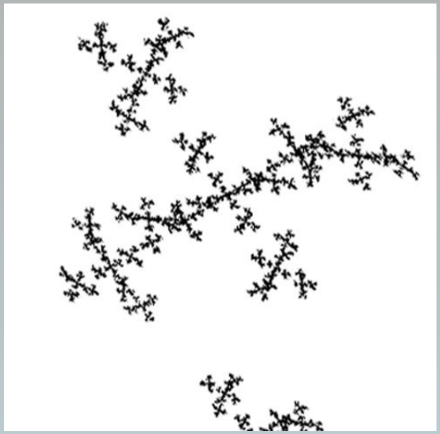


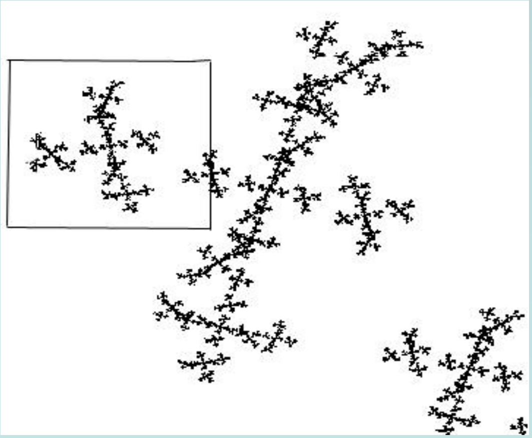


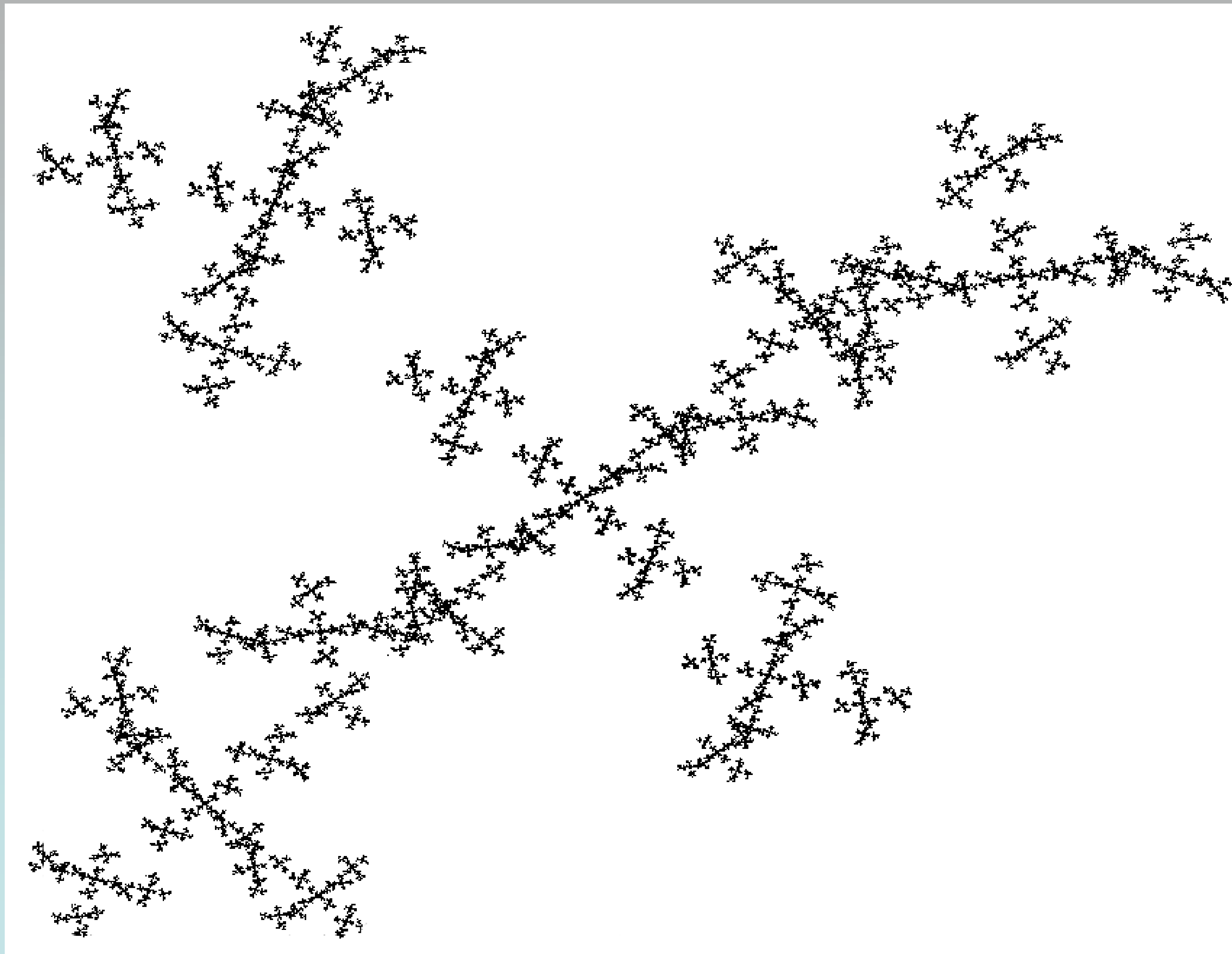
Self-Similarity



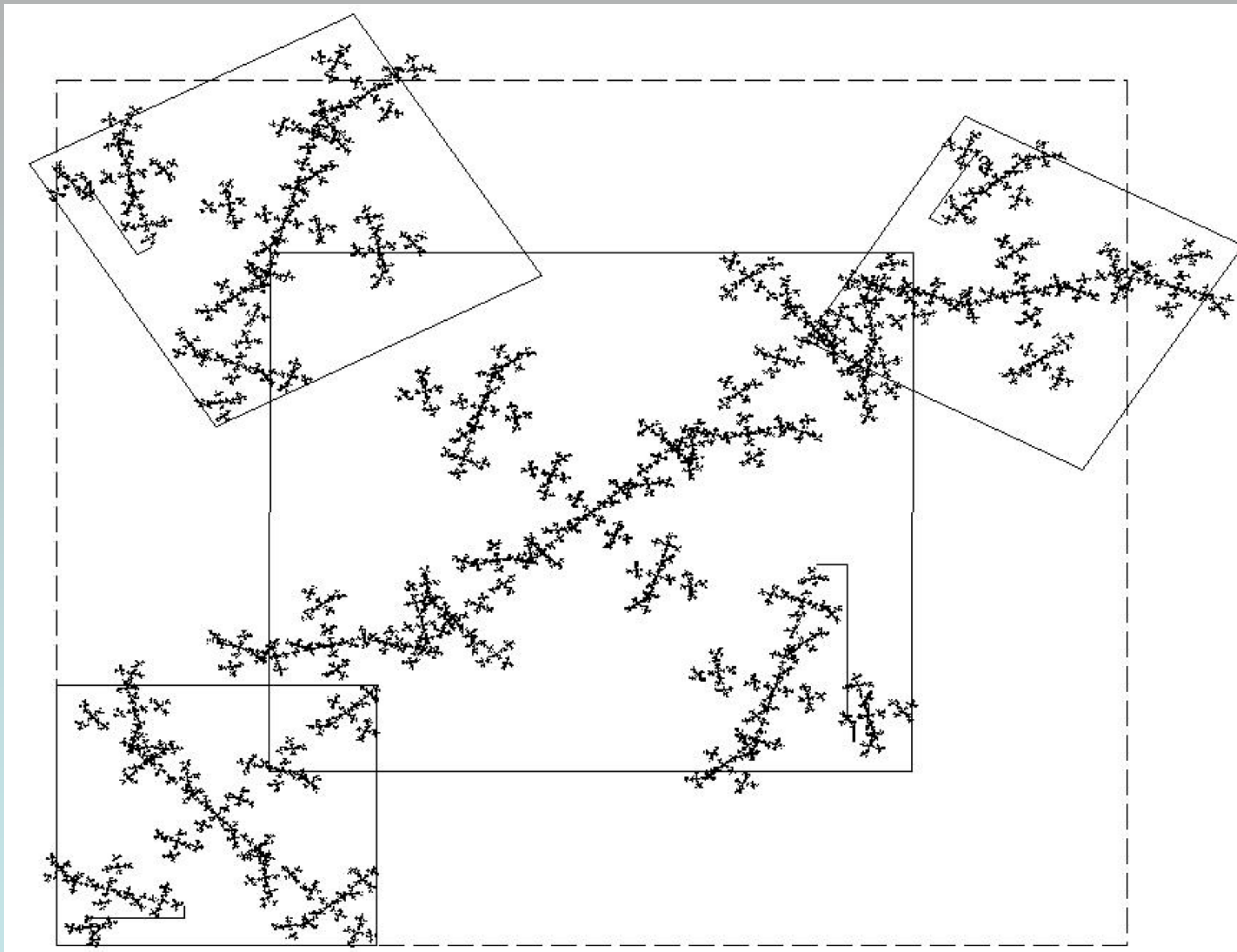


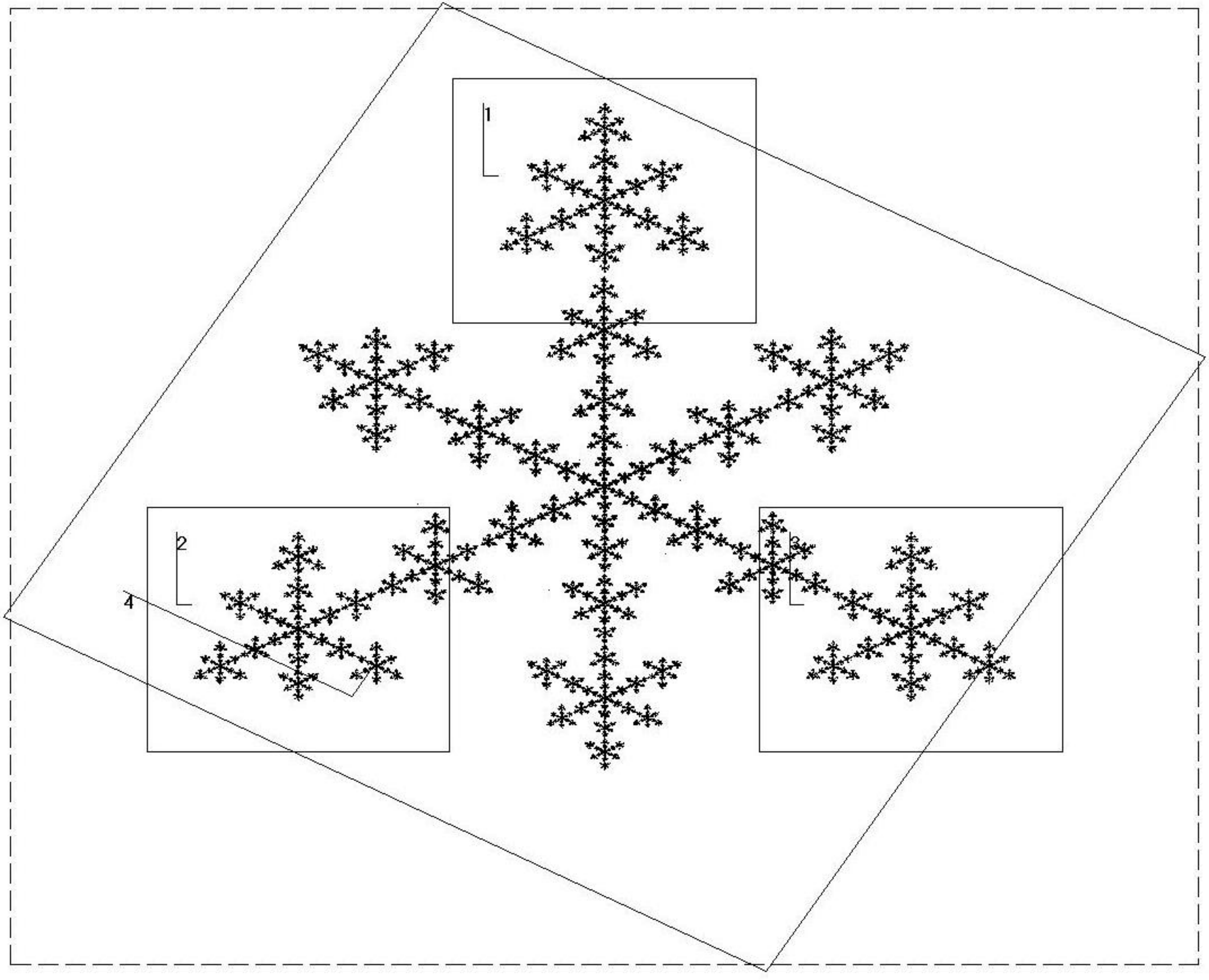




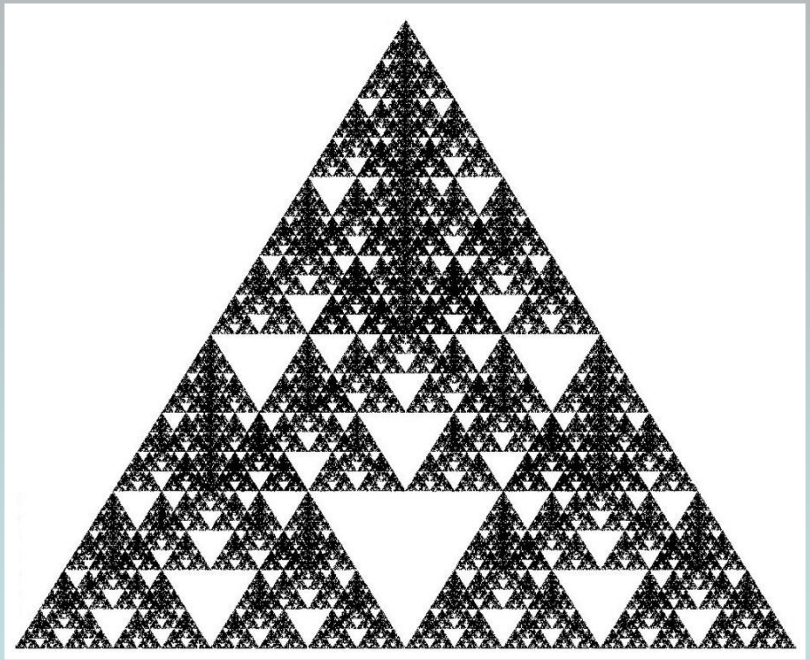


Four self-similar pieces → Infinitely many self-similar pieces

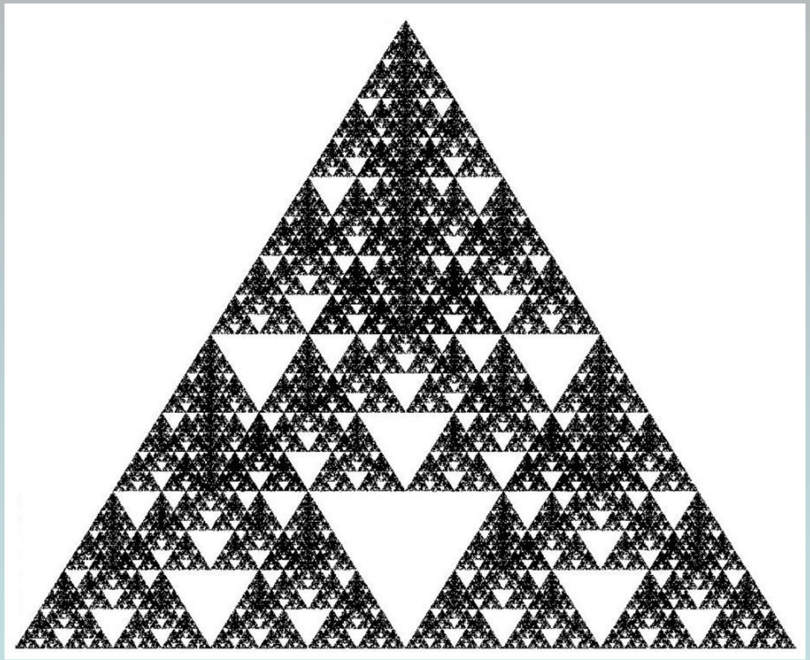



1
2
3
4

Is this a fractal? (self-similar?)

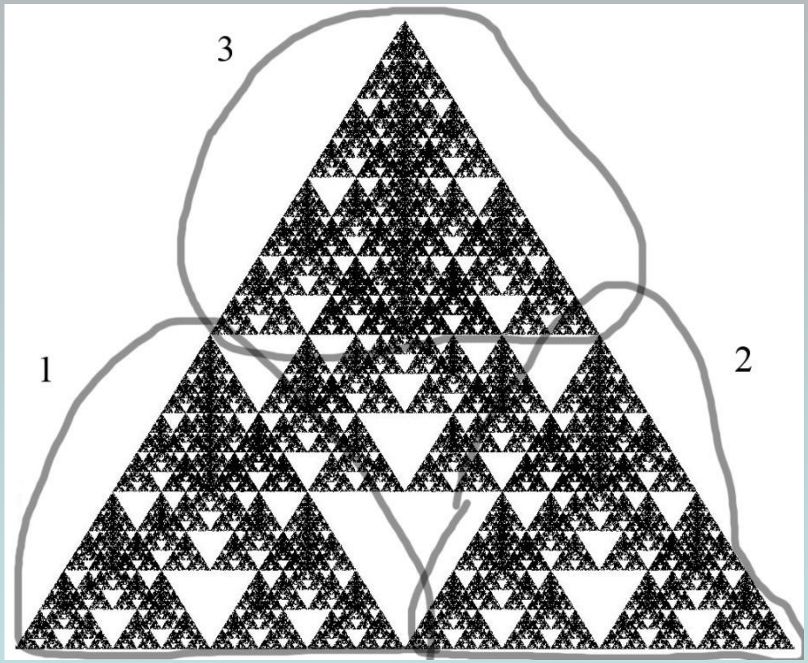


Is this a fractal? (self-similar?)



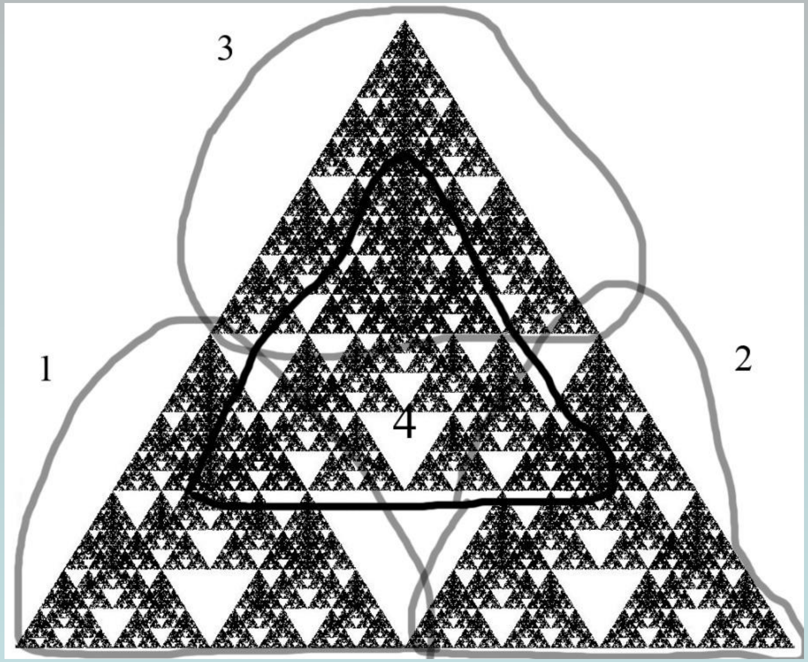
Yes – 4 pieces, but there is overlap

Is this a fractal? (self-similar?)



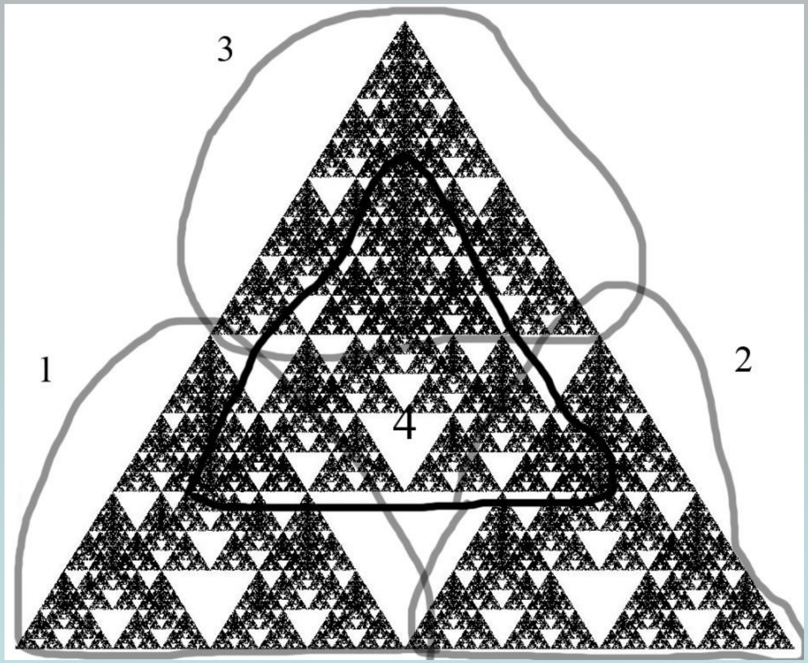
Yes – 4 pieces, but there is overlap

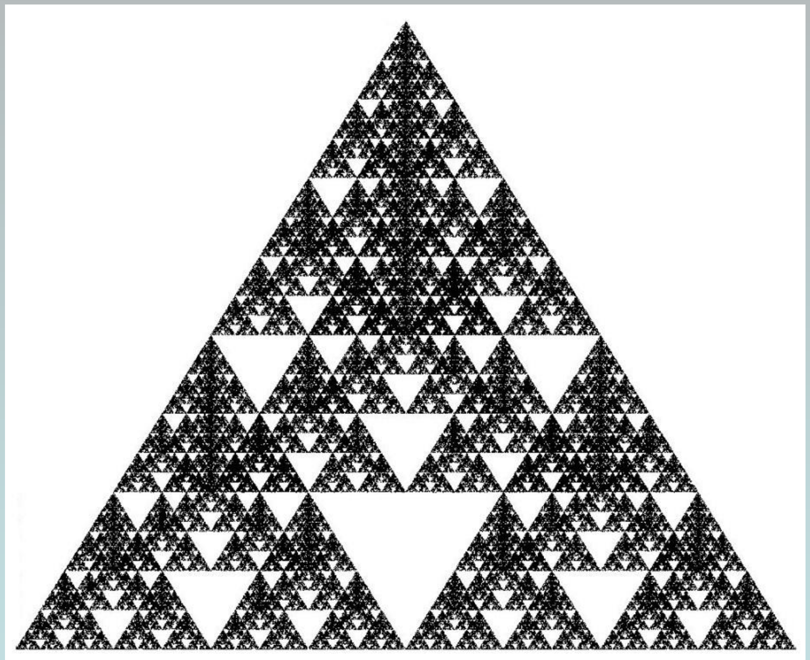
Is this a fractal? (self-similar?)

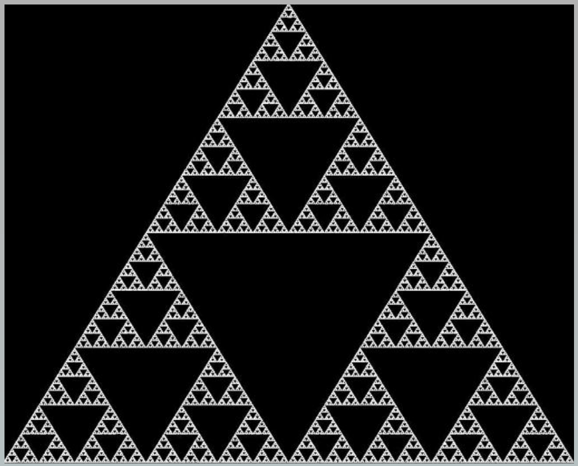
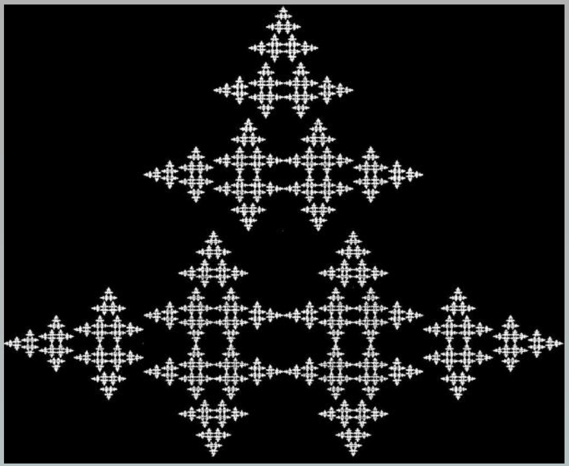
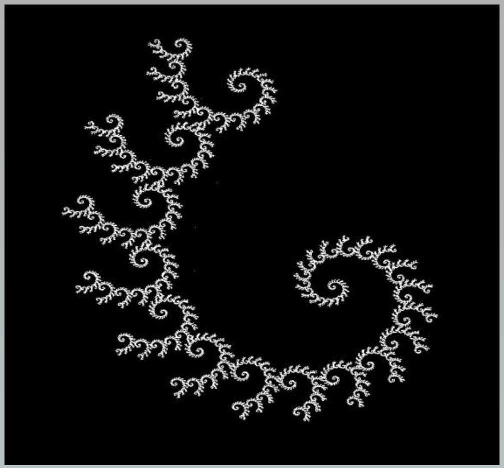
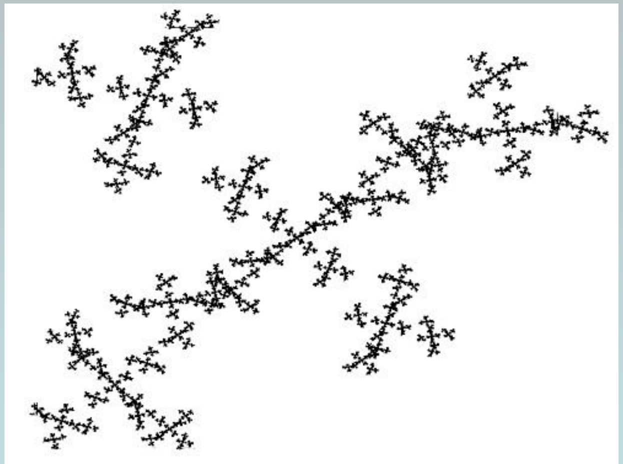


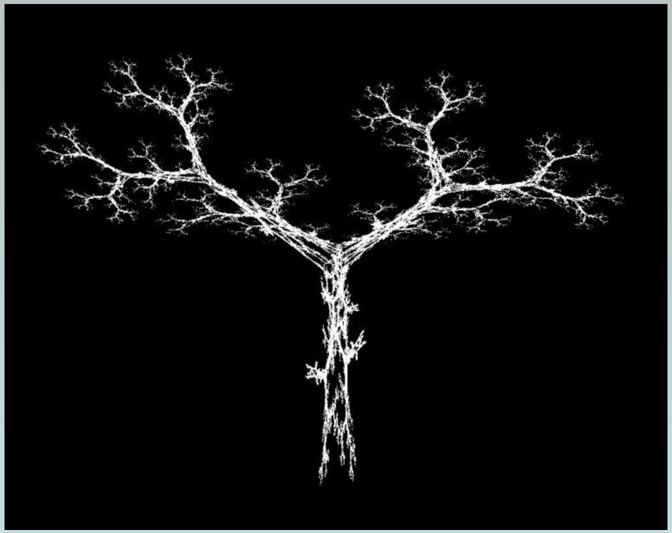
Yes – 4 pieces, but there is overlap

4 self-similar pieces





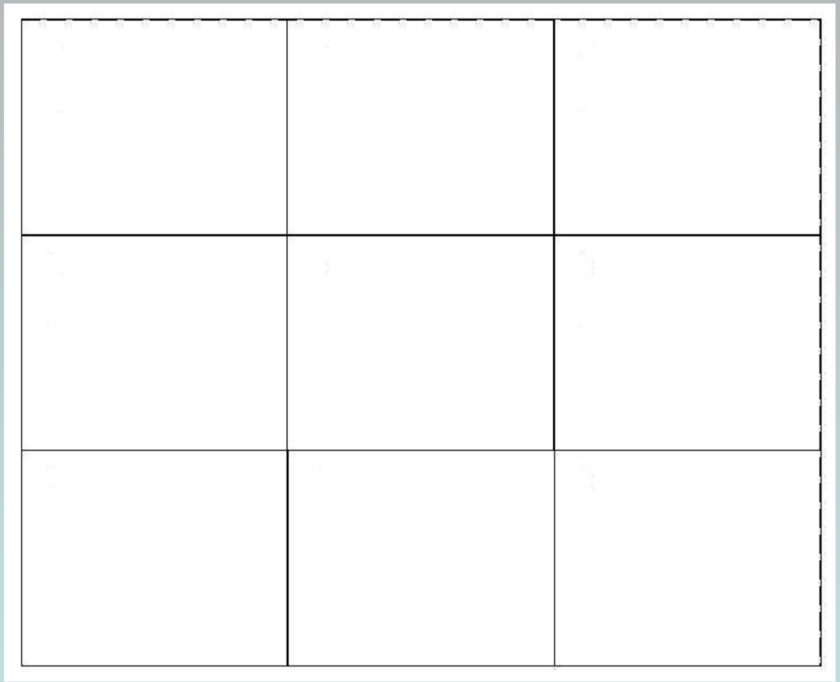







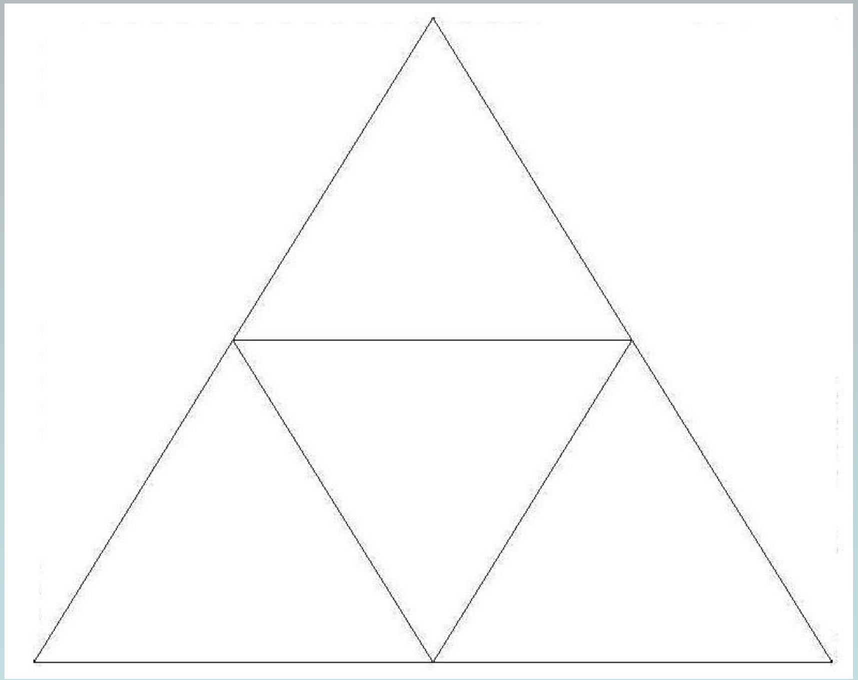
Self-similar : Made up of smaller copies of itself

Other self-similar objects:

Solid square



Solid triangle

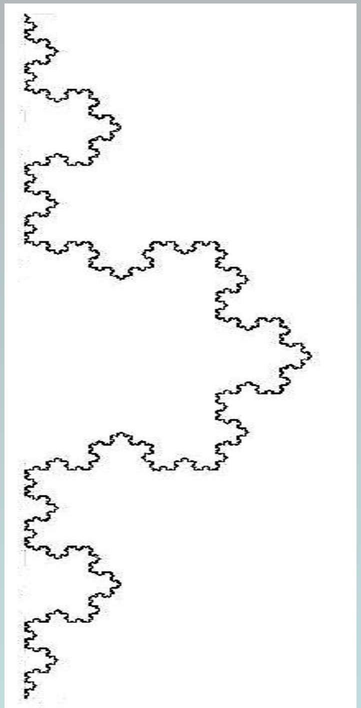


These are self-similar, but not “complicated”

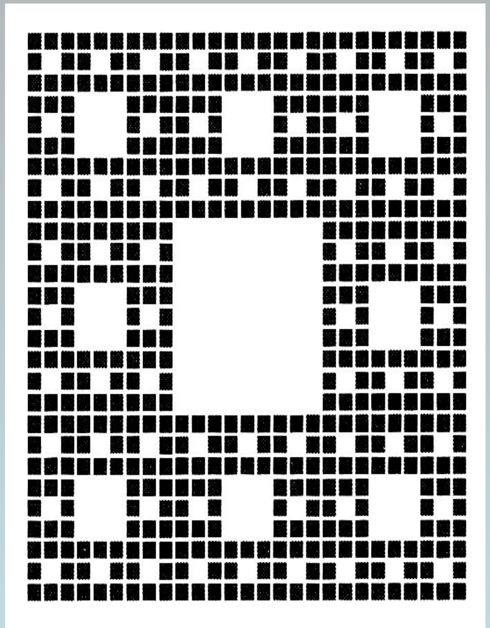
Fractals are self-similar <u>and</u> complicated

Fractal dust

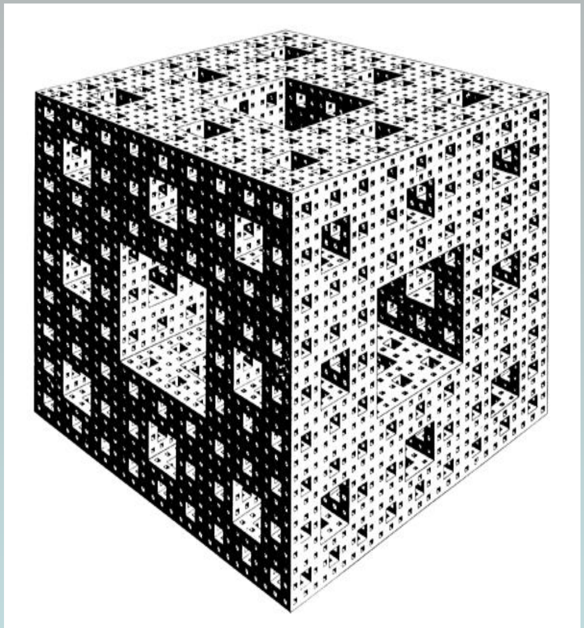
Fractal curves



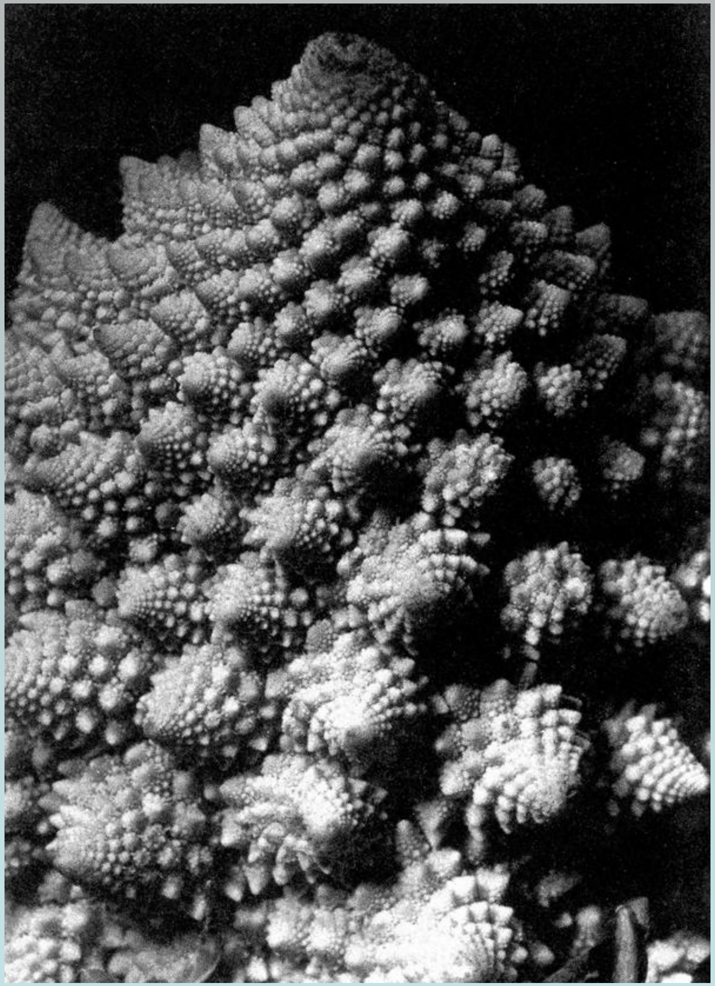
Fractal areas

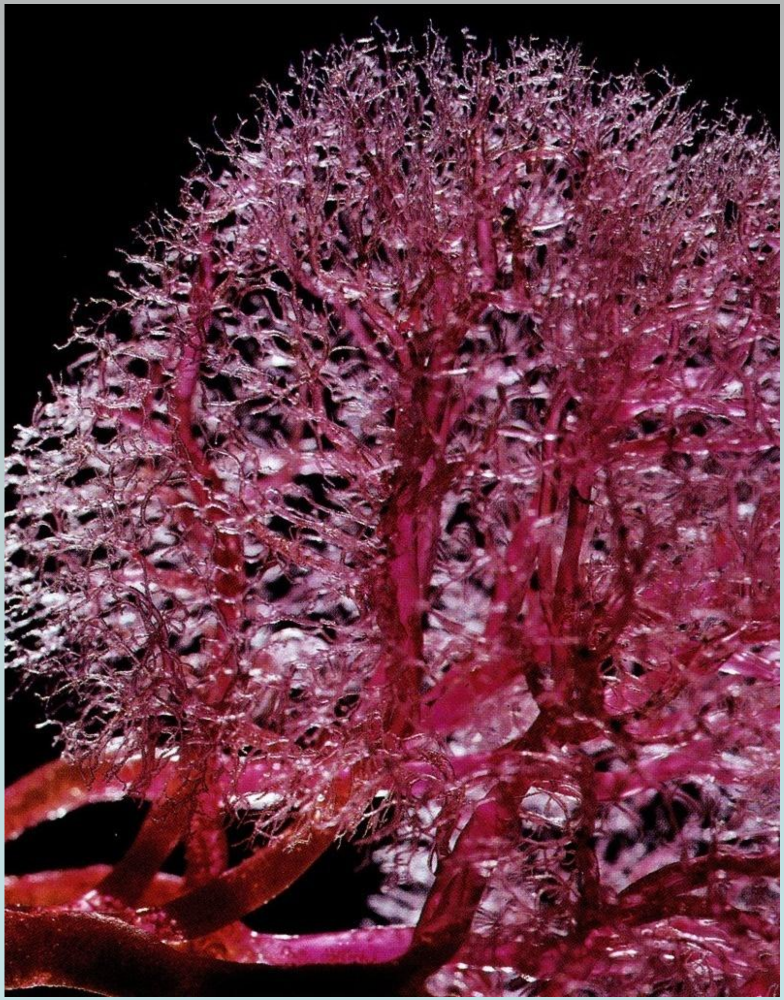


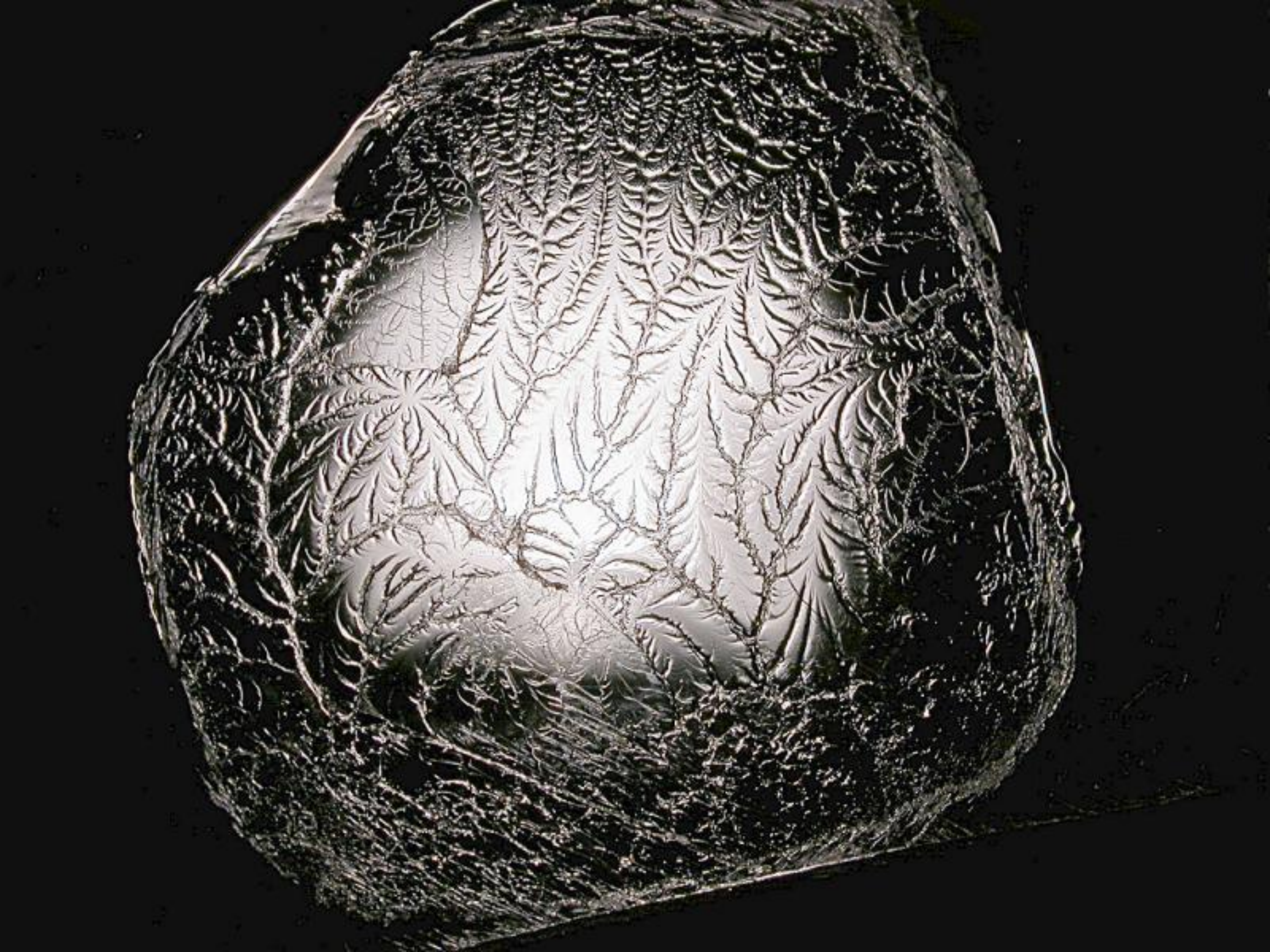
Fractal volumes

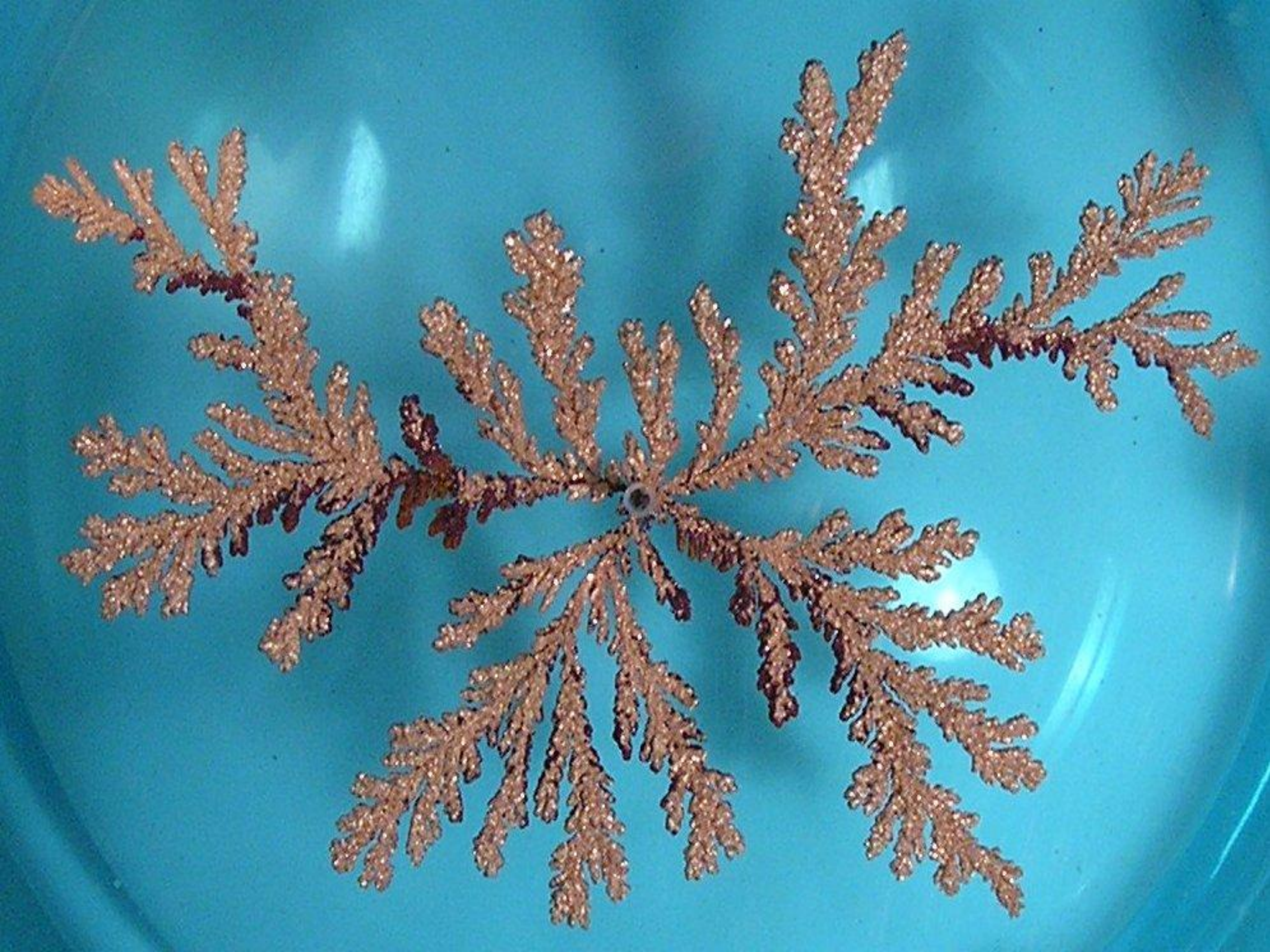


Fractals in Nature





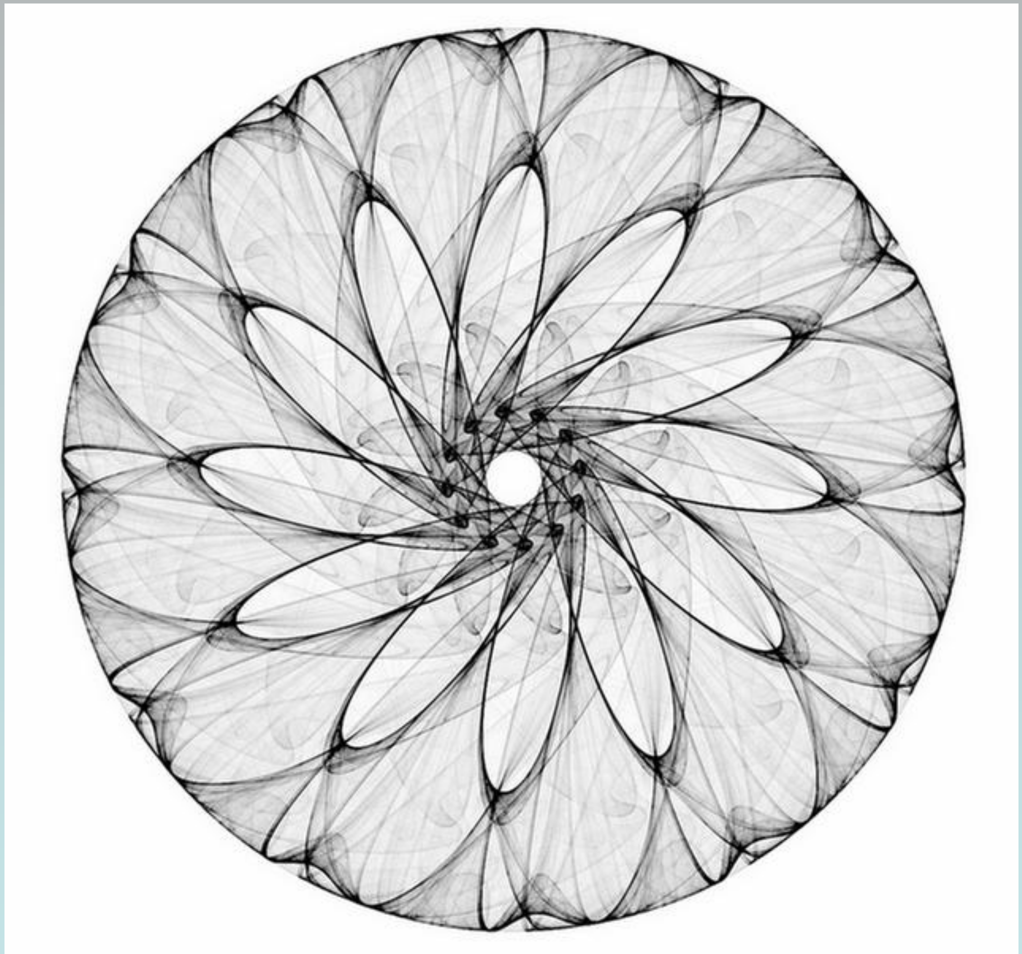



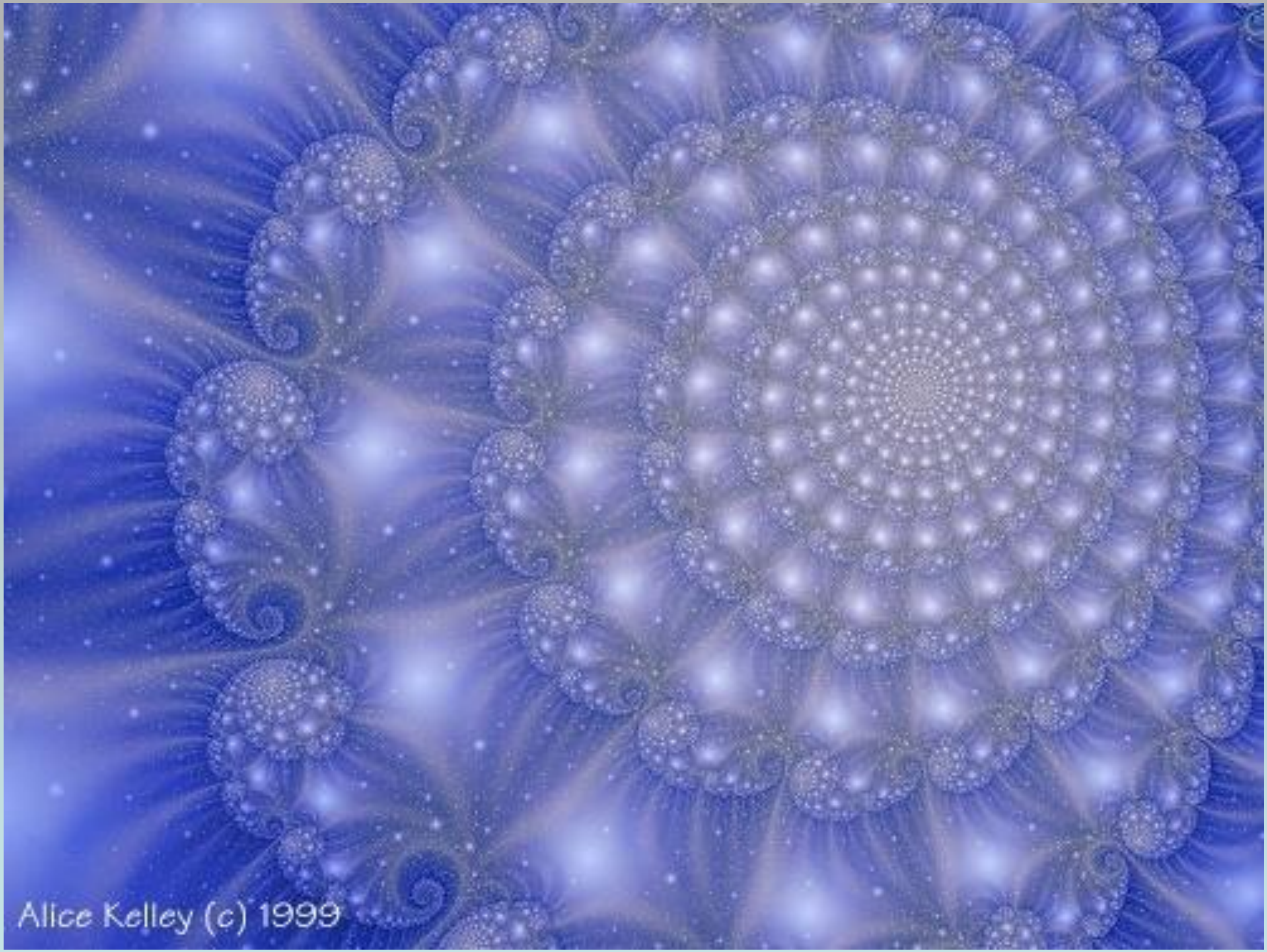


Real Mountains

Fractal Mountains

Fractal Art

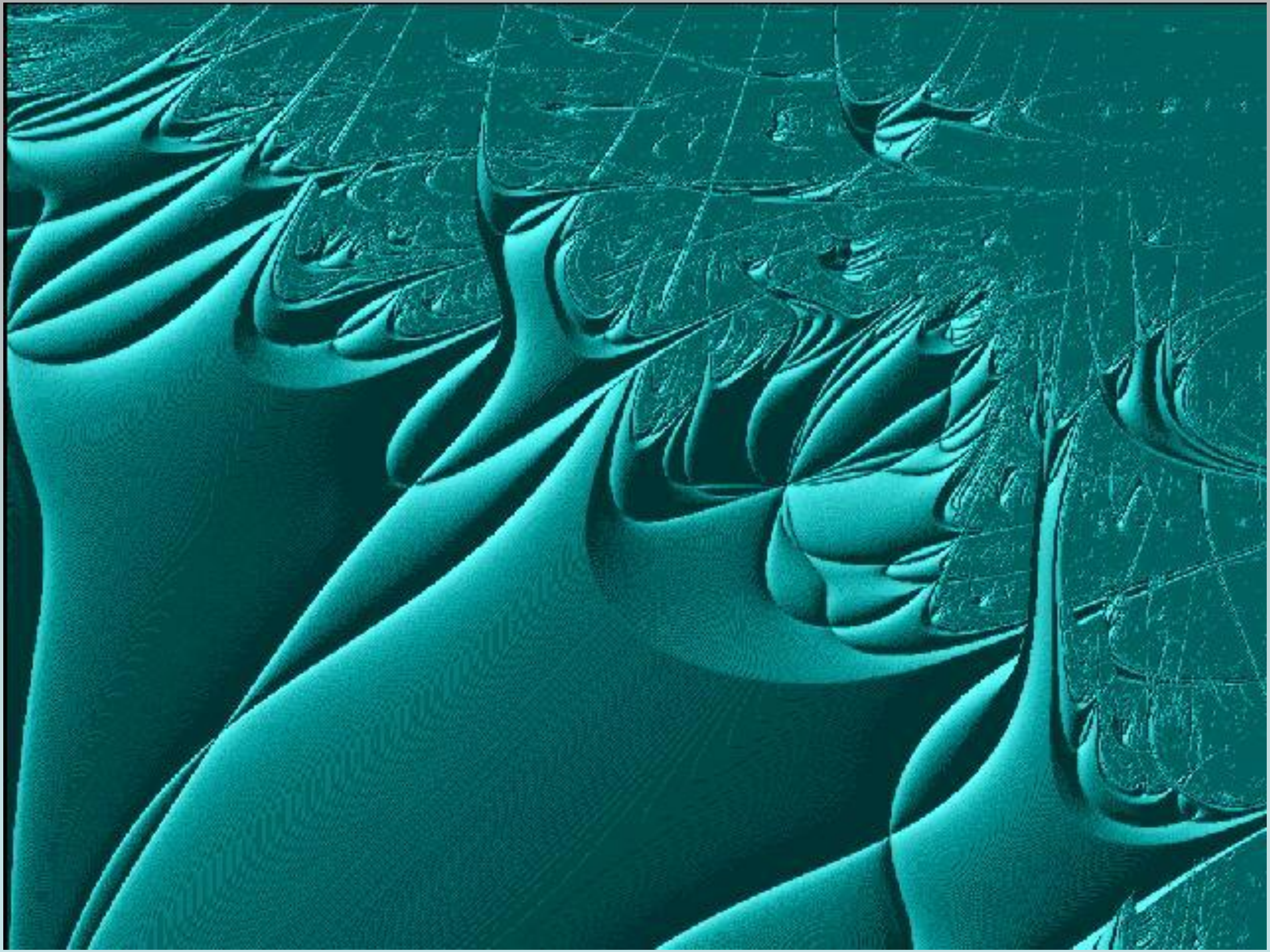



Alice Kelley (c) 1999


Alice Kelley (c) 1999


BAILEAU.III
JLMG2006 - FRACTOVIA.ORG


Alice Kelley (c) 1999




Alice Kelley (c) 1999

How do we draw fractals?

Drawing fractals

- Here are four ways:

  - Iteration of an IFS (‘decoding’)

Drawing fractals

- Here are four ways:
  - Iteration of an IFS
  - Removing pieces

Drawing fractals

- Here are four ways:

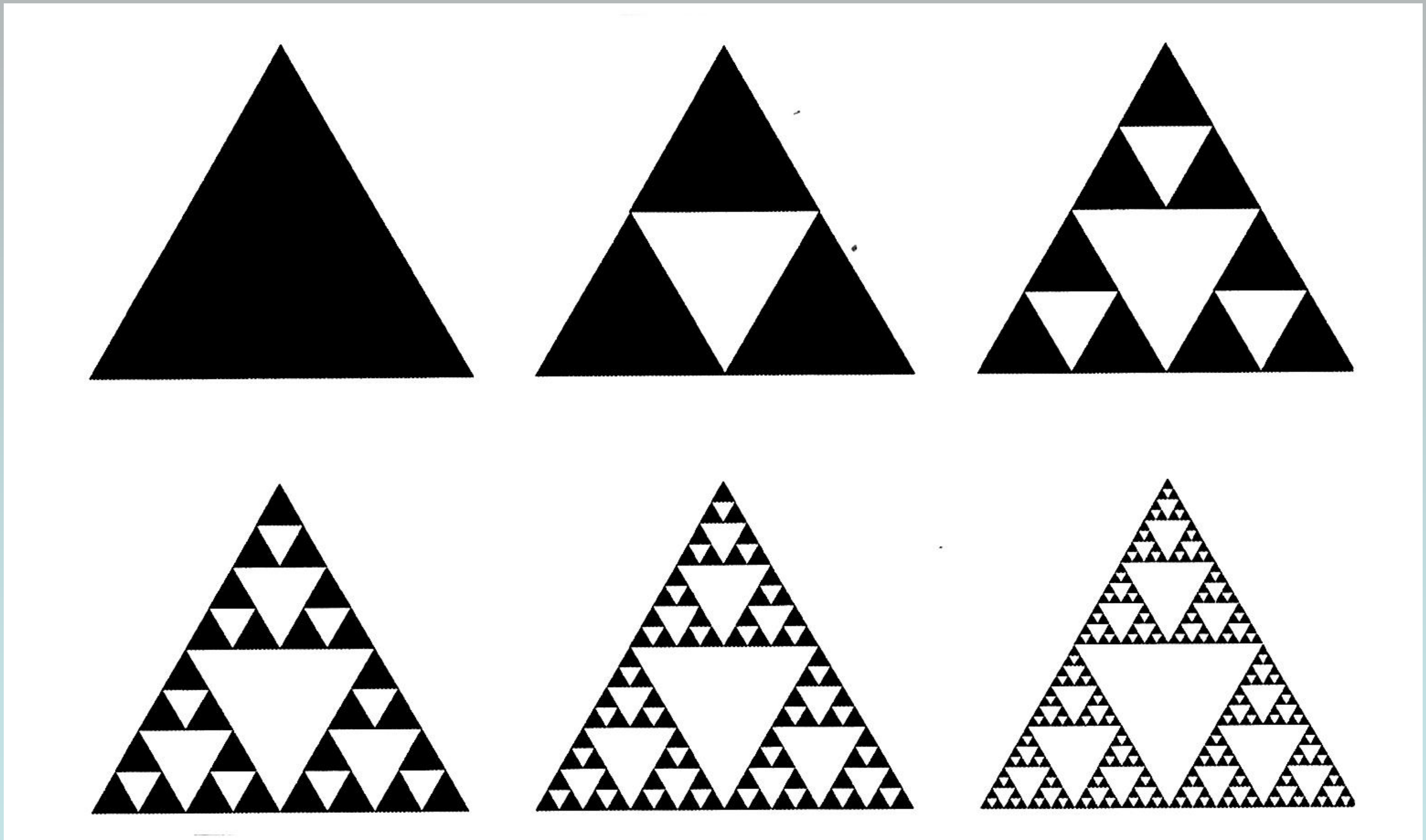
  - Iteration of an IFS
  - Removing pieces
  - Adding pieces

Drawing fractals

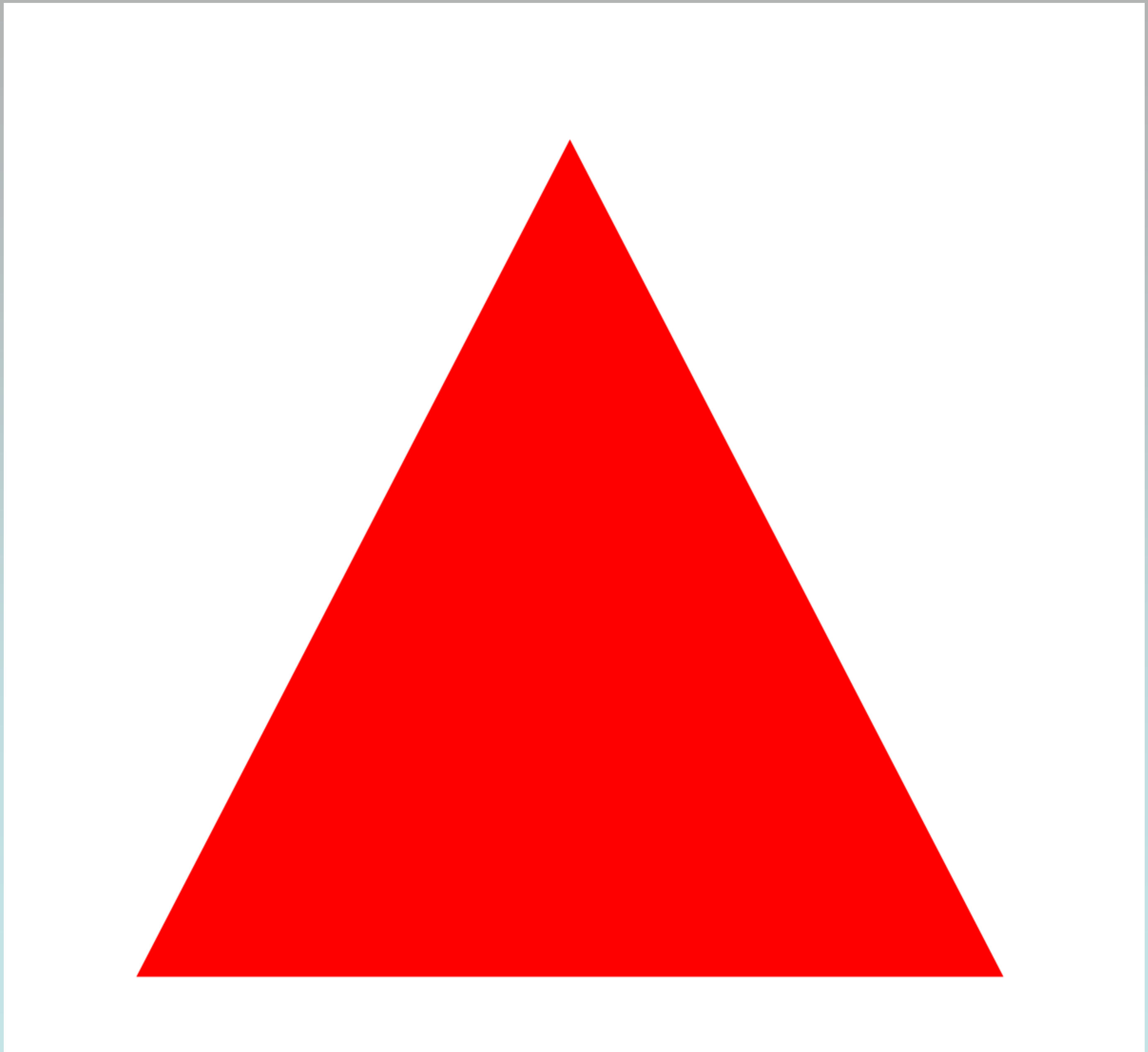
- Here are four ways:

  - Iteration of an IFS
  - Removing pieces
  - Adding pieces
  - The ‘Chaos Game’

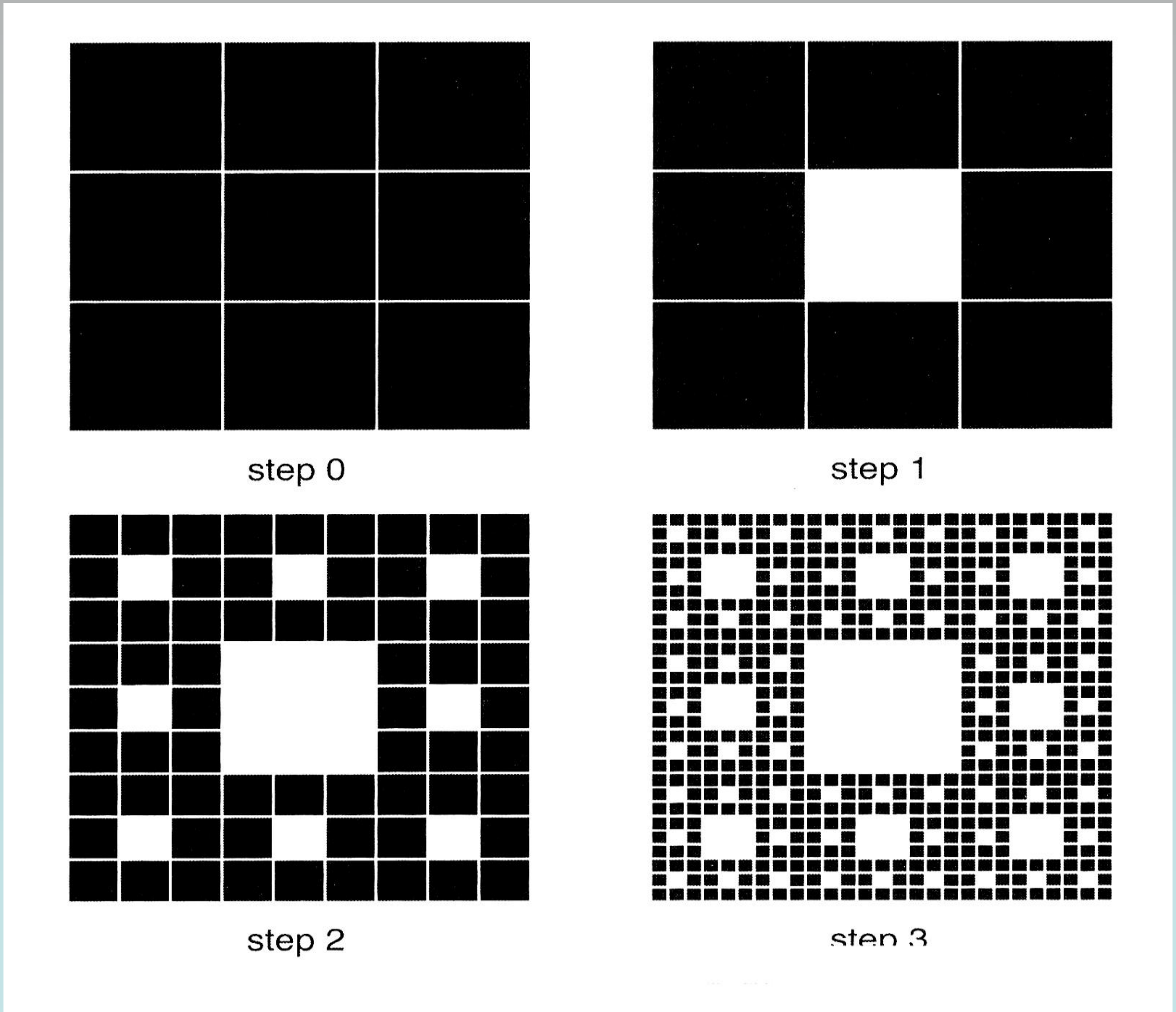
Removing pieces



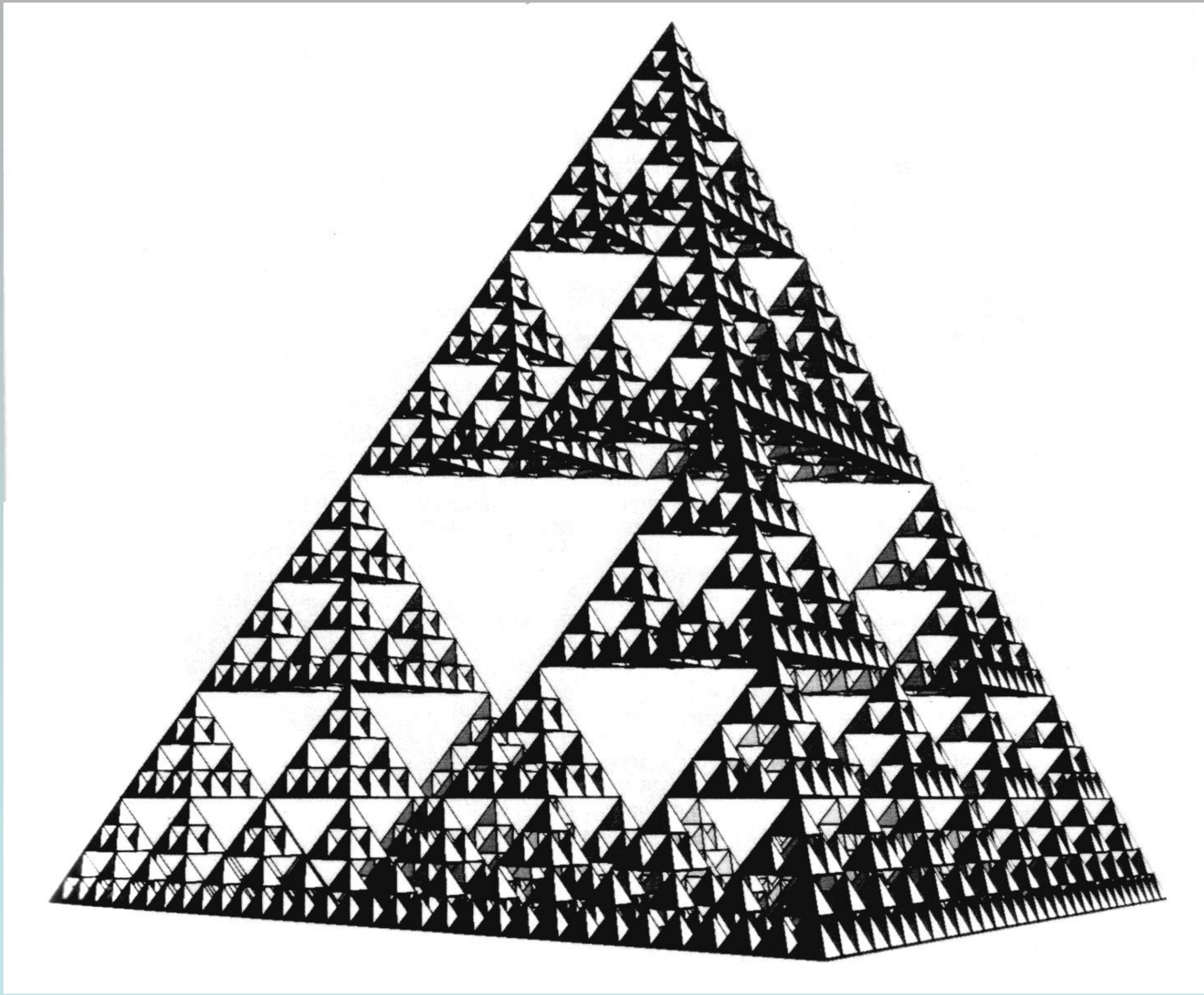
Wikipedia; 'fractals'

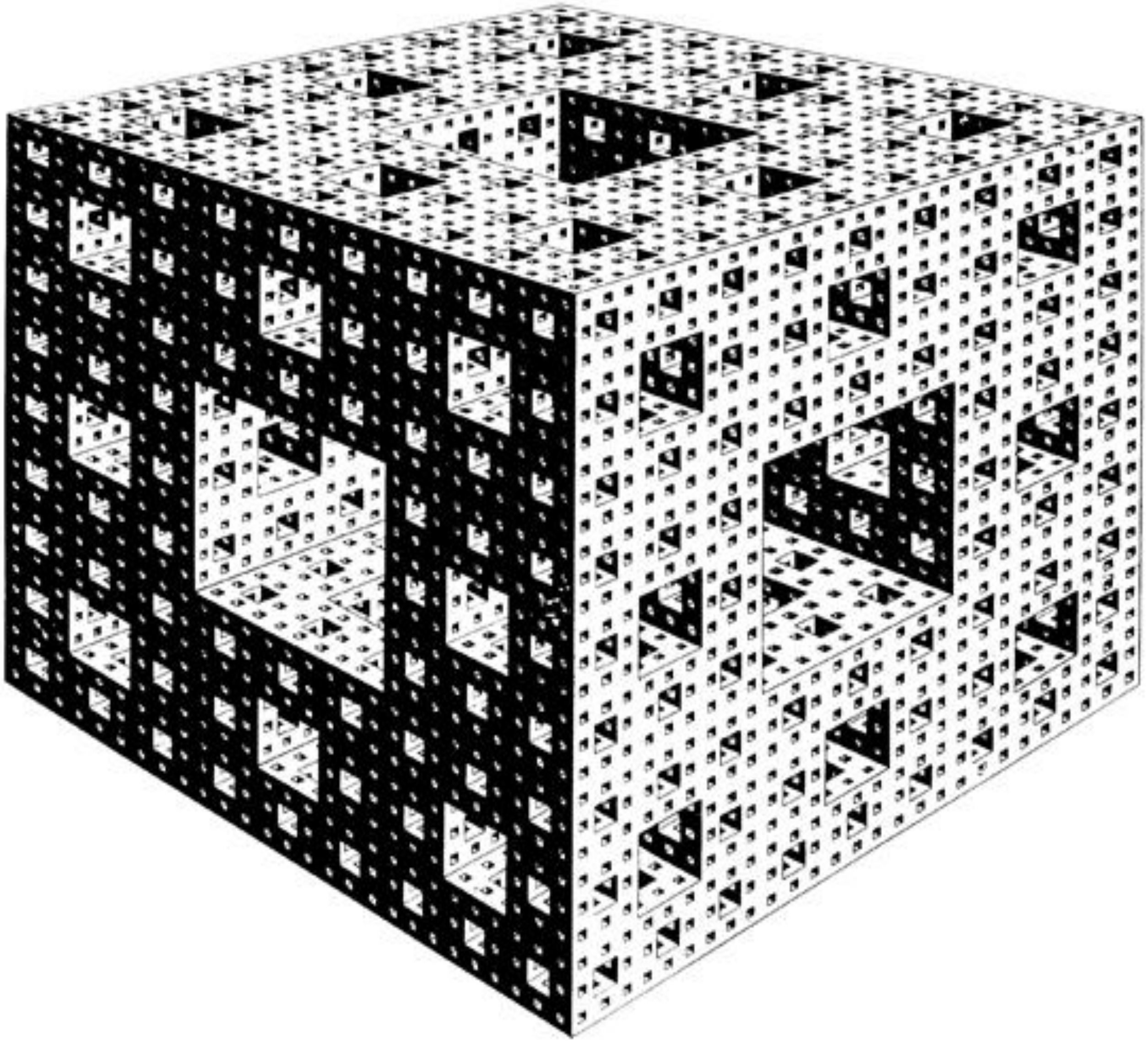


Removing pieces



Or in 3-D

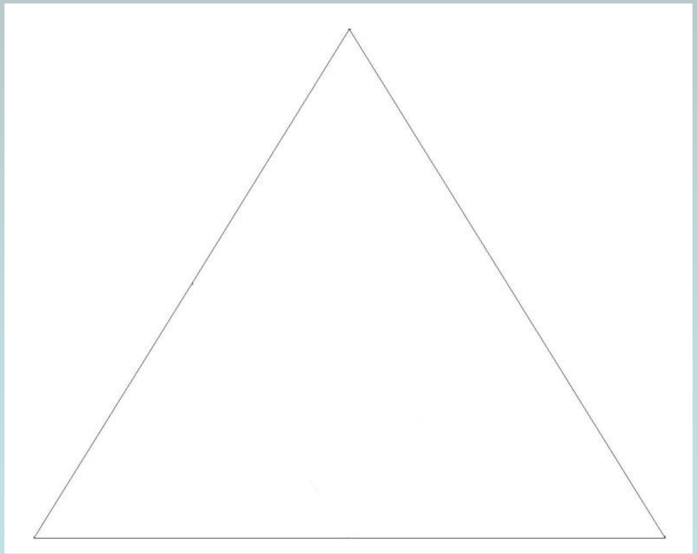




Now you try

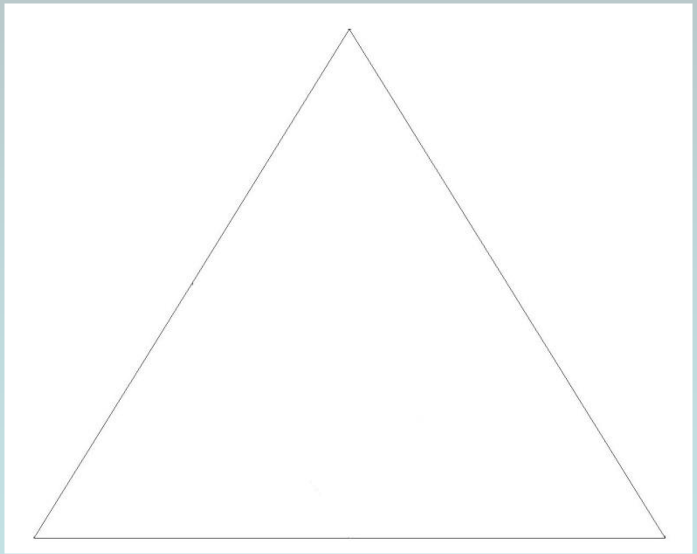
Now you try

Start with a triangle:

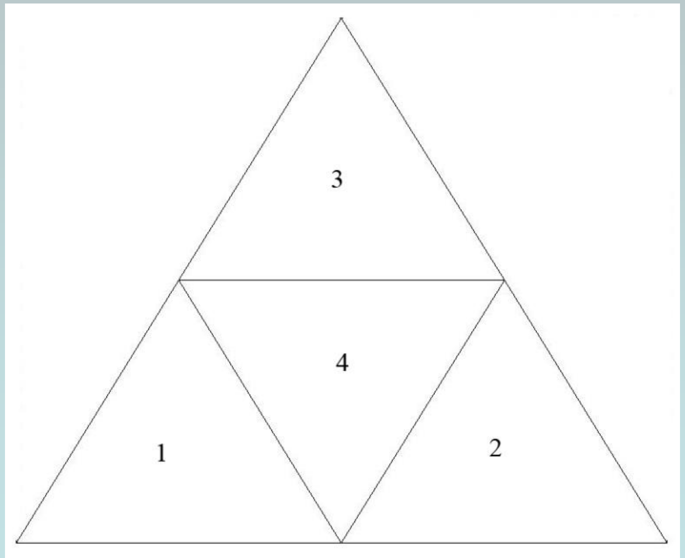


Now you try

Start with a triangle:

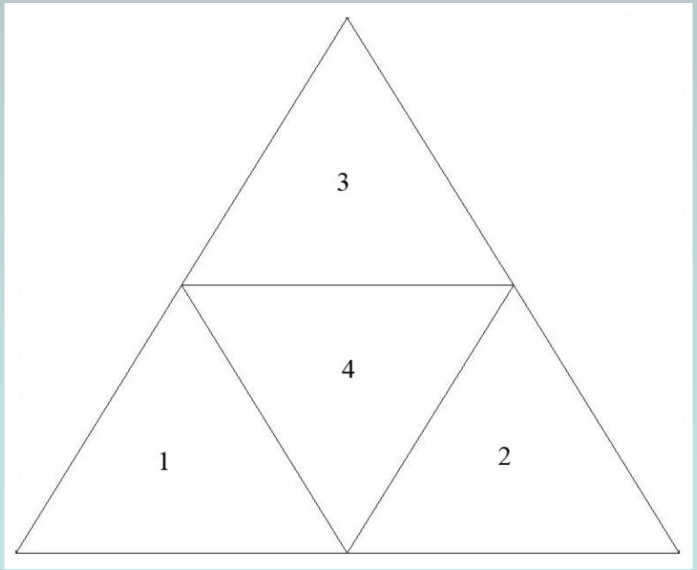


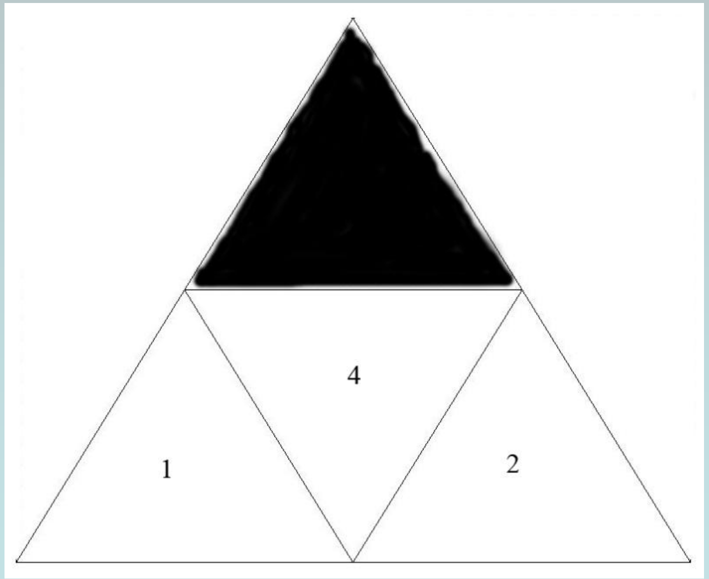
Partition it into 4 smaller triangles:



Now you try

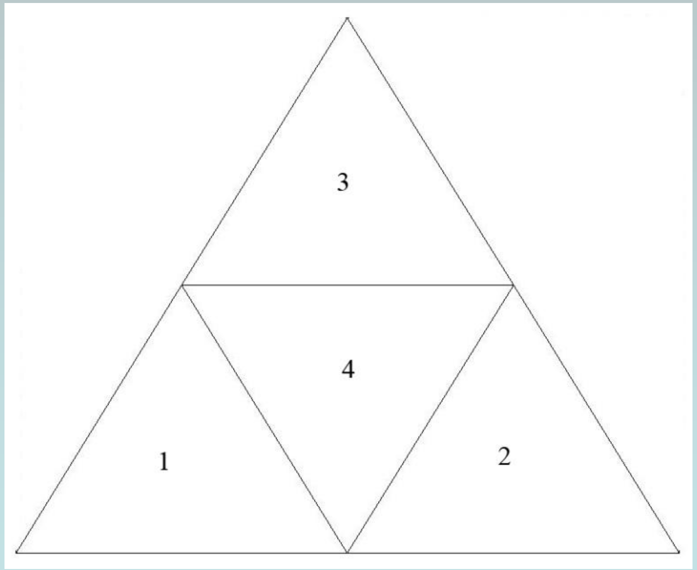
Remove sub-triangle 3:

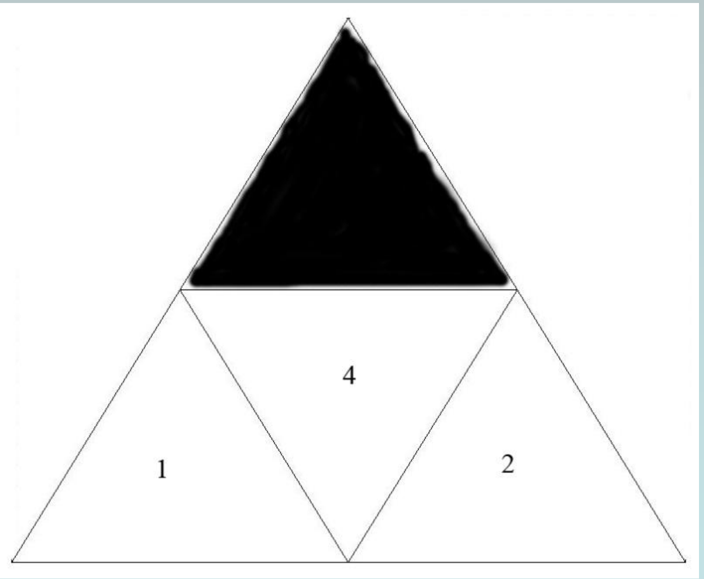




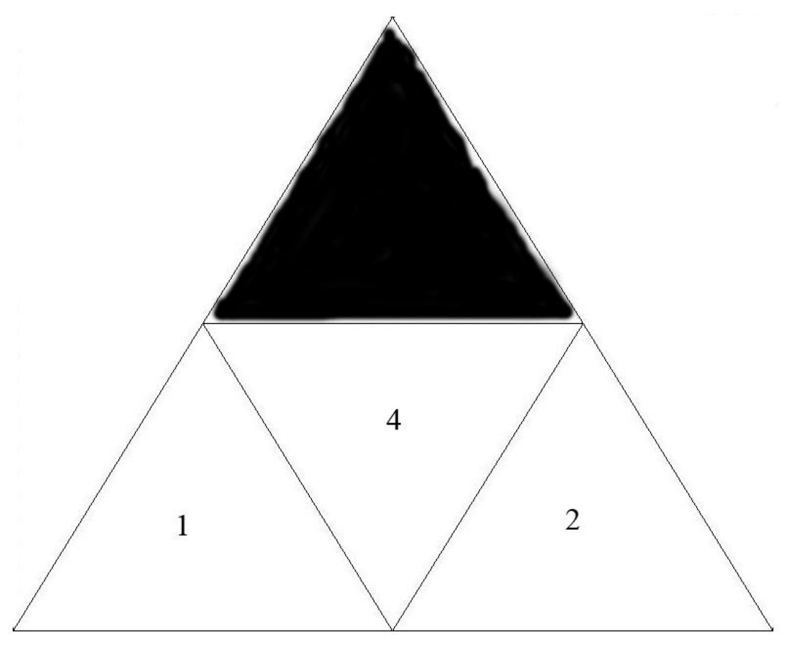
Now you try

Remove sub-triangle 3:

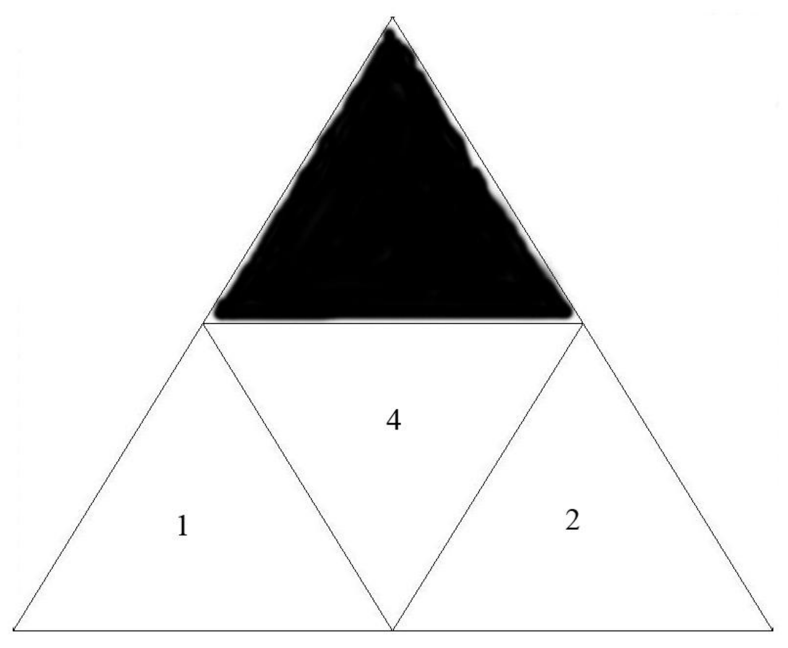


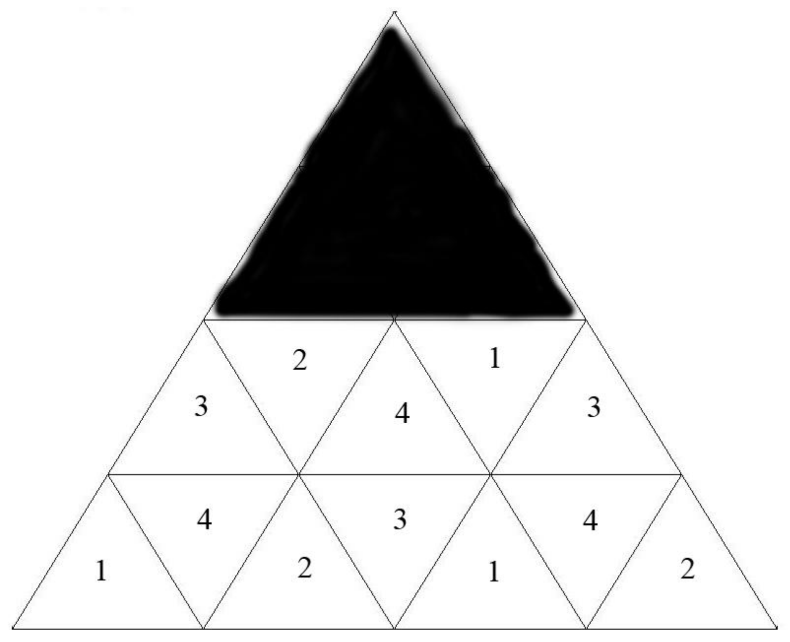


Now continue . . .

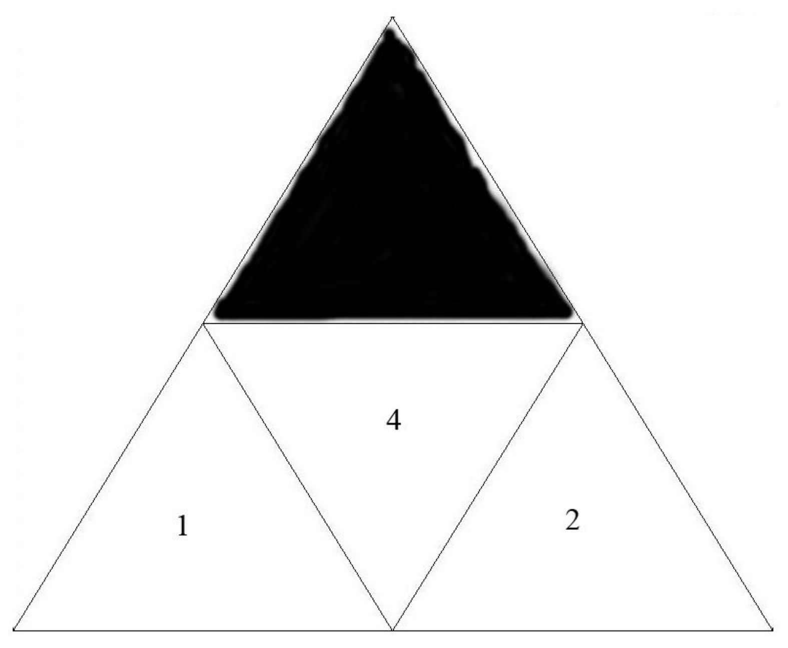


Partition each remaining triangle
into 4 smaller triangles

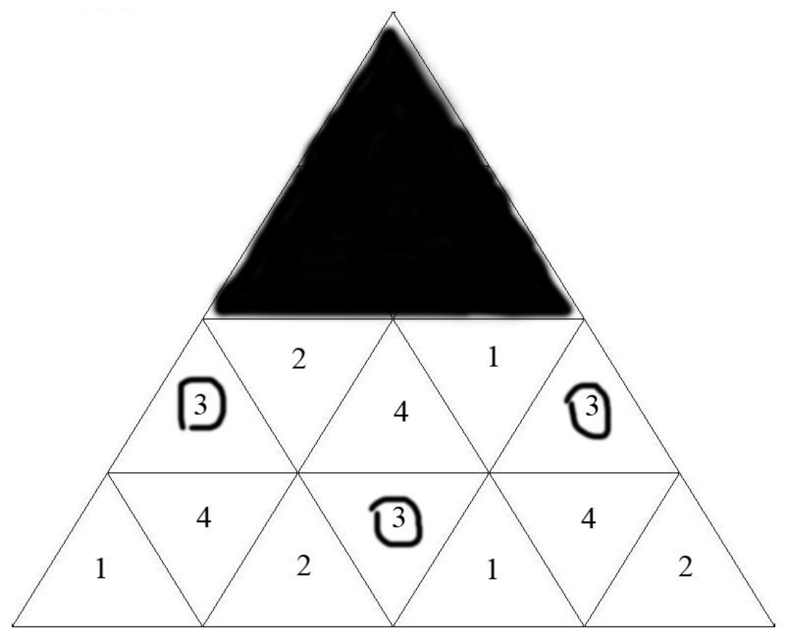




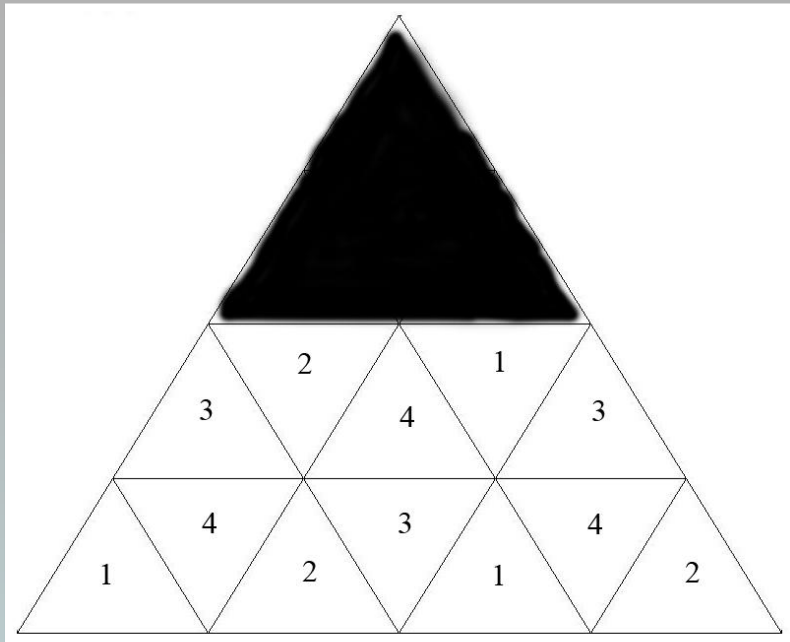
Partition each remaining triangle
into 4 smaller triangles

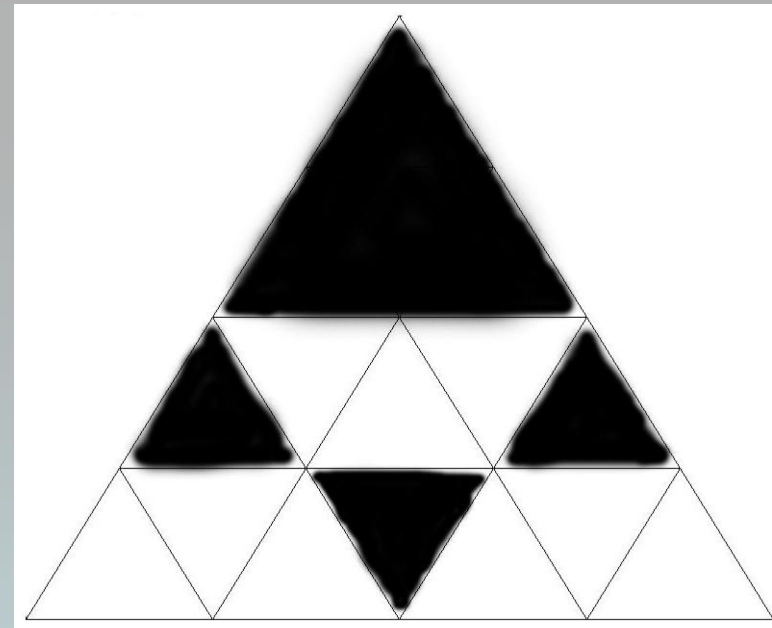


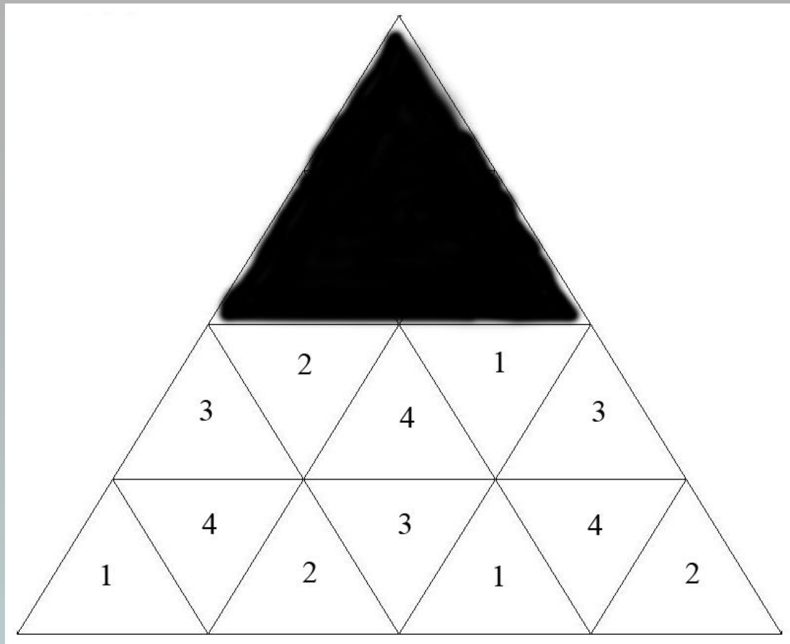
Partition each remaining triangle
into 4 smaller triangles

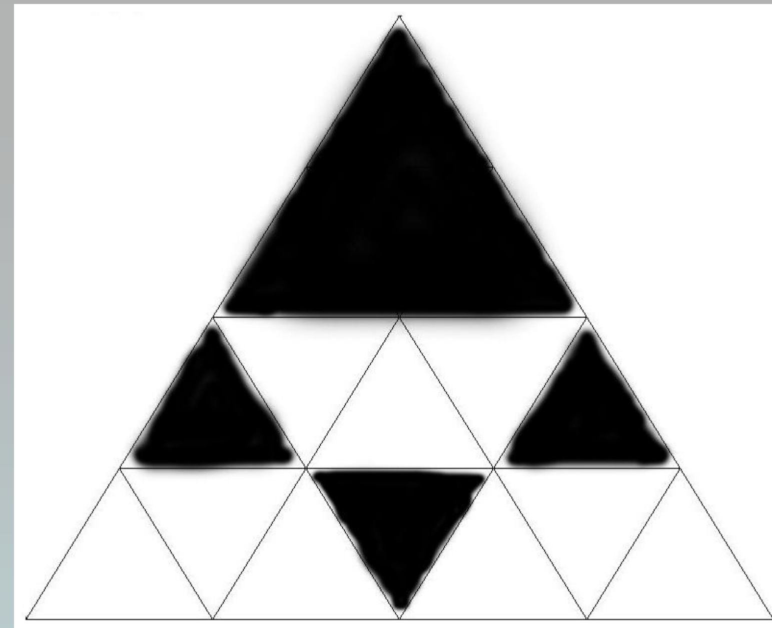


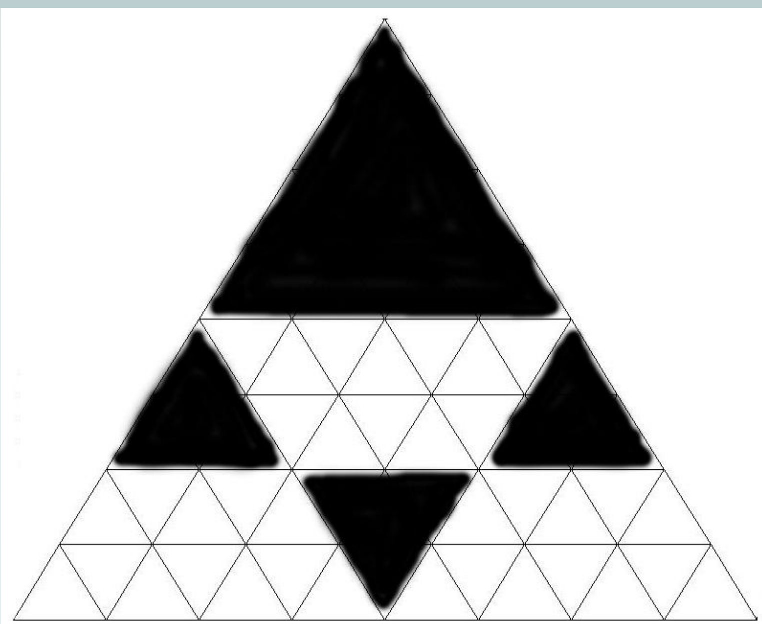
Now remove each ‘3’ triangle

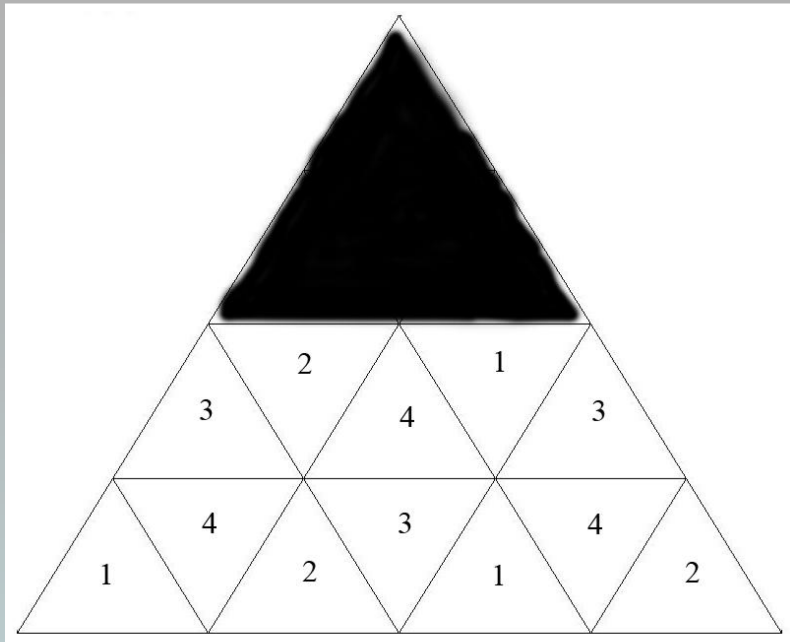

2
1
3
4
3
4
3
4
1
2
1
2

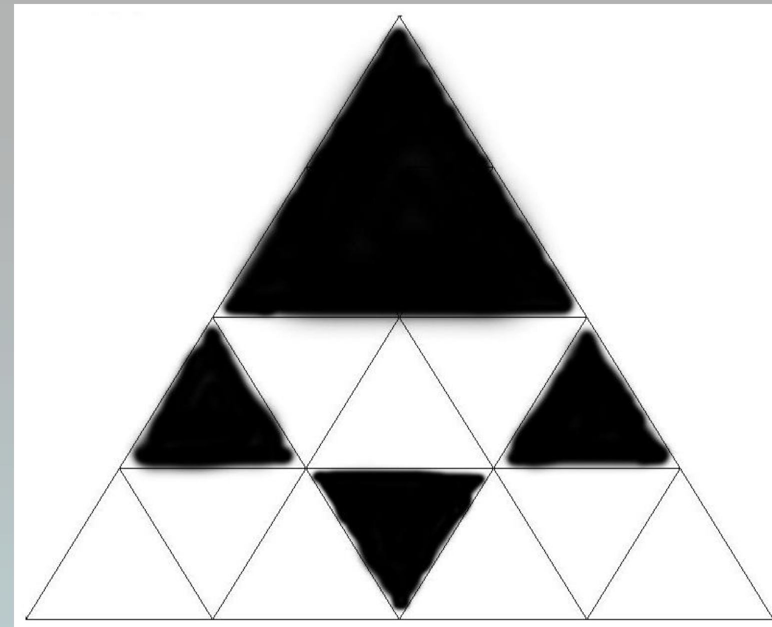


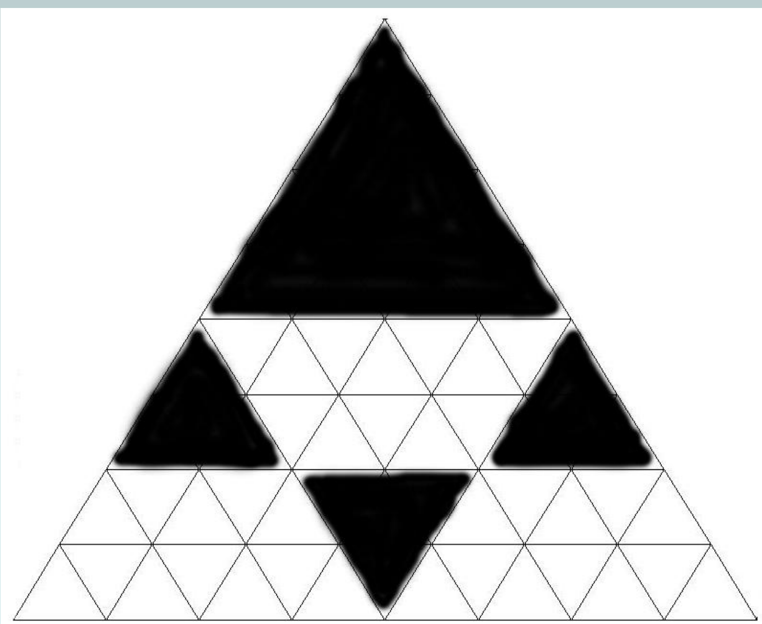

2
1
3
4
3
4
3
4
1
2
1
2

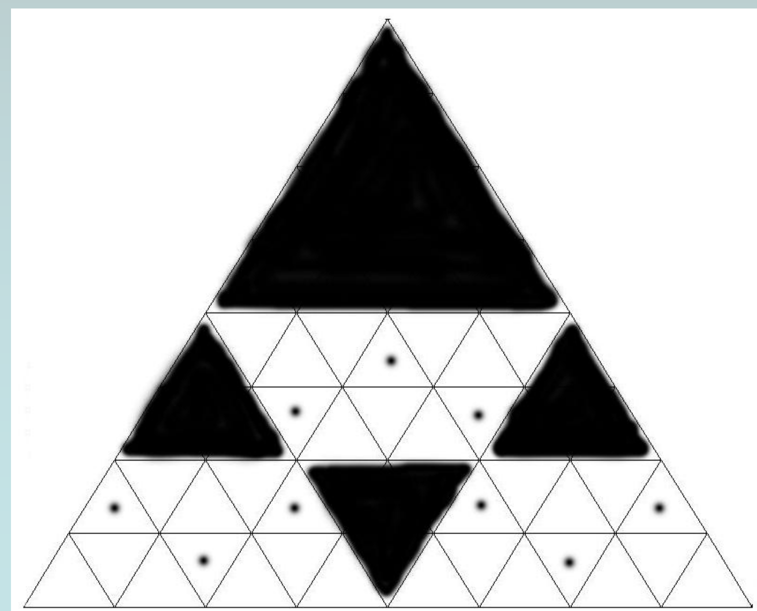


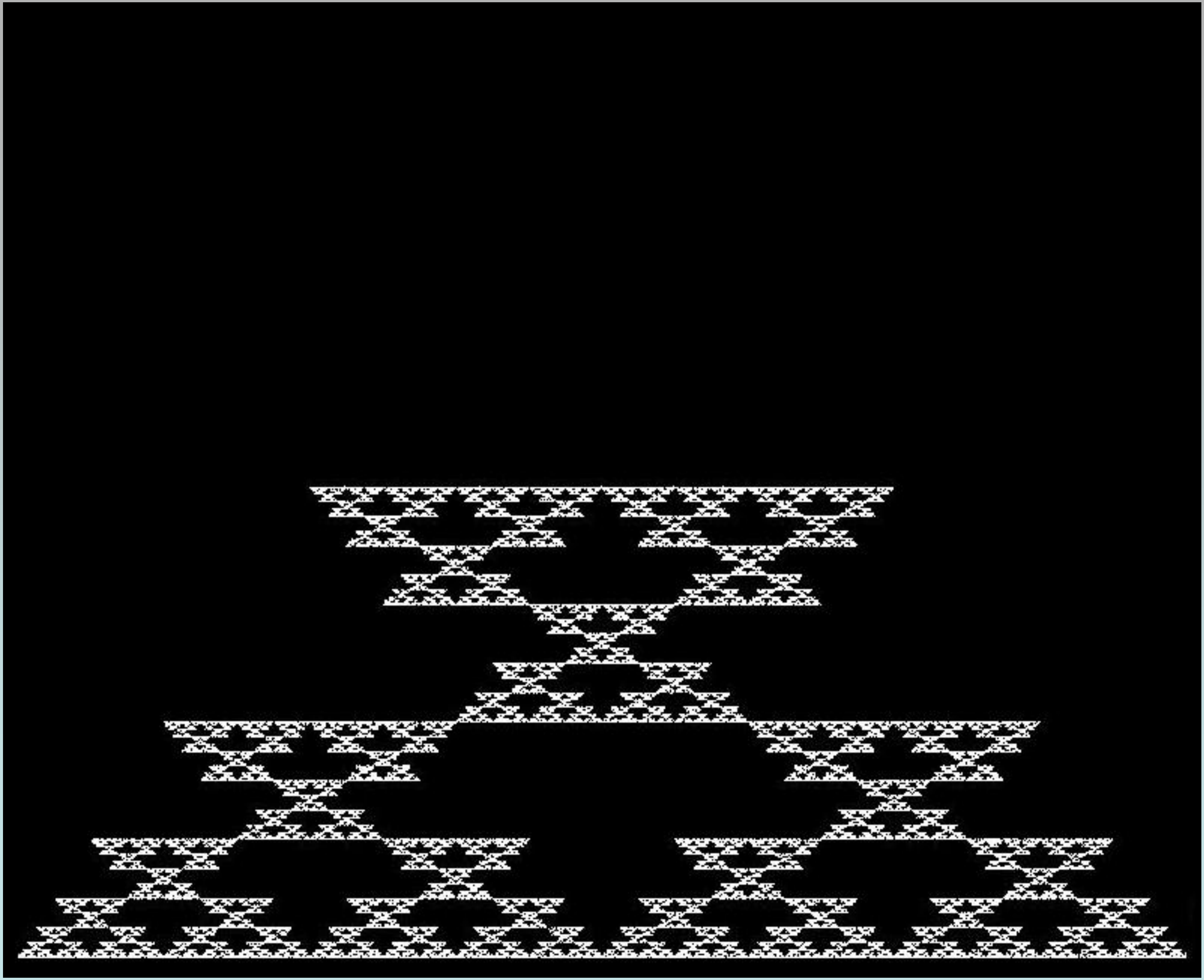


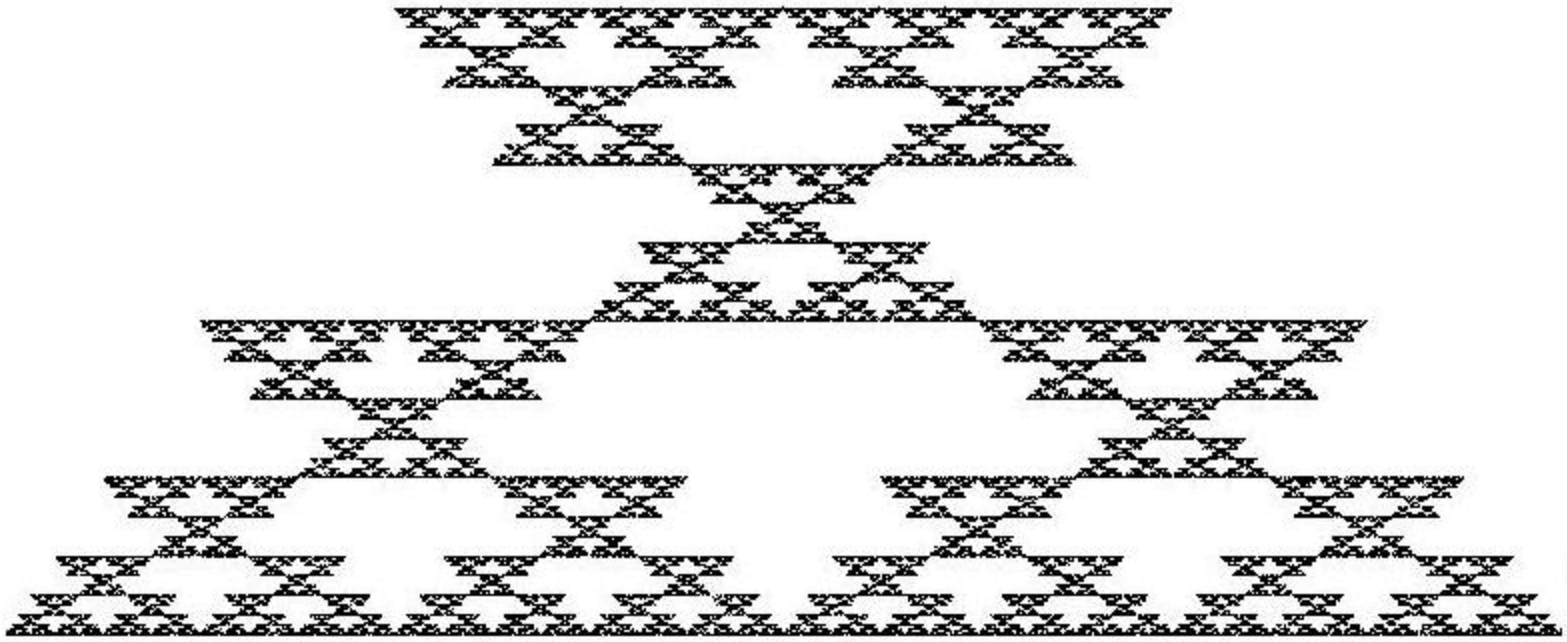

2 1
3 4 3
4 3 4
1 2 1 2



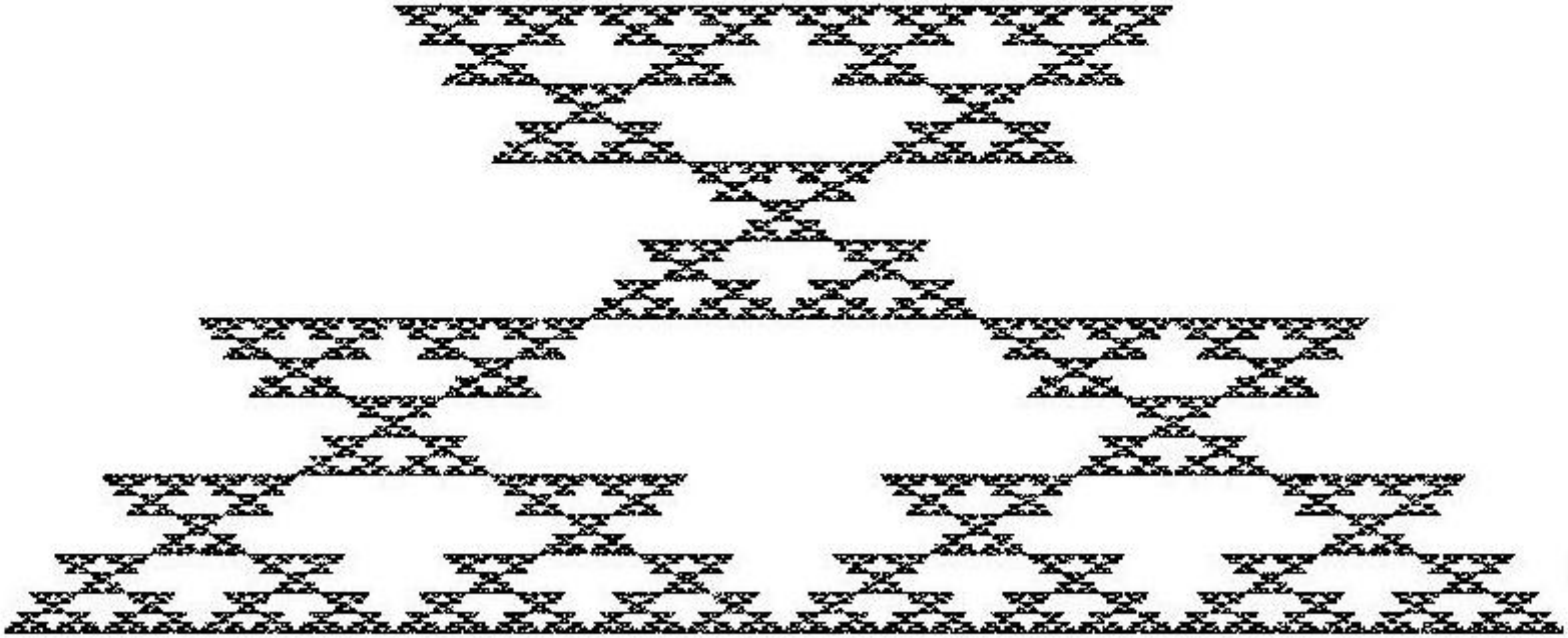




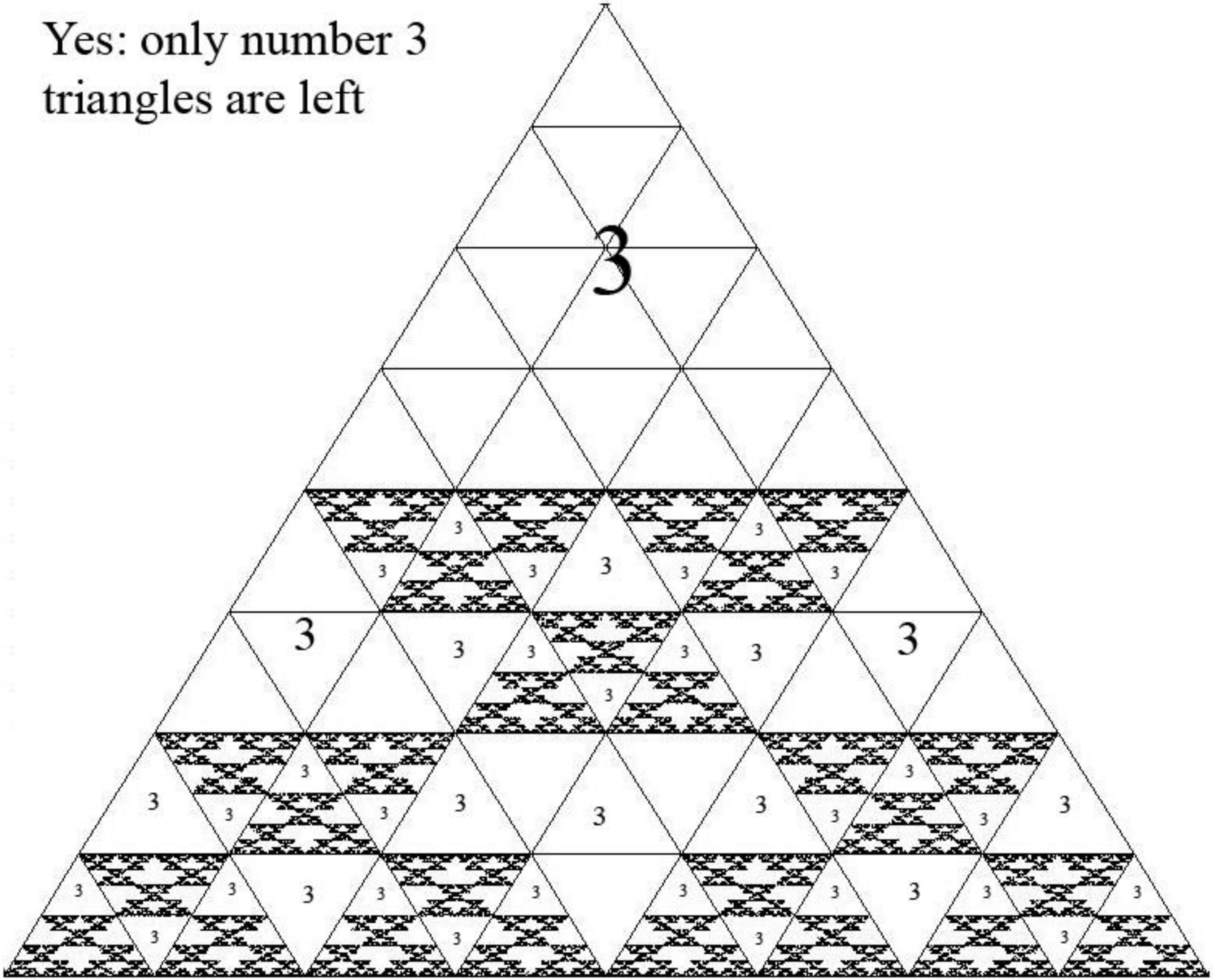




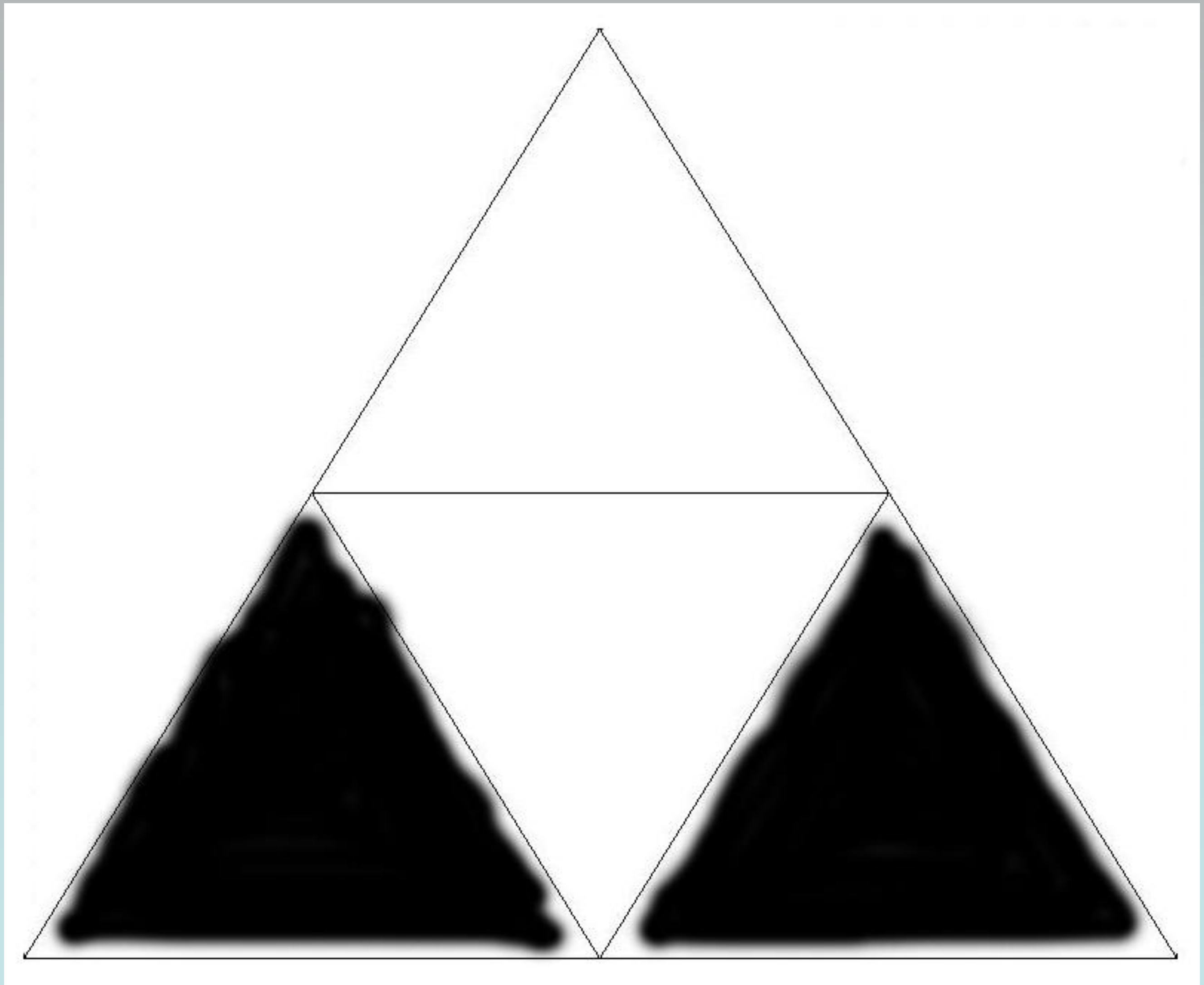
But how do you know this is the result?

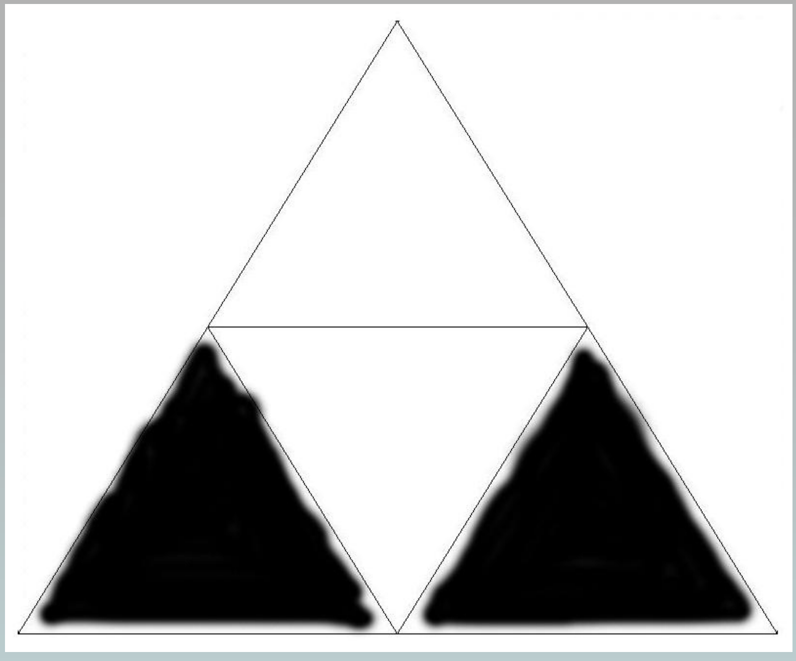


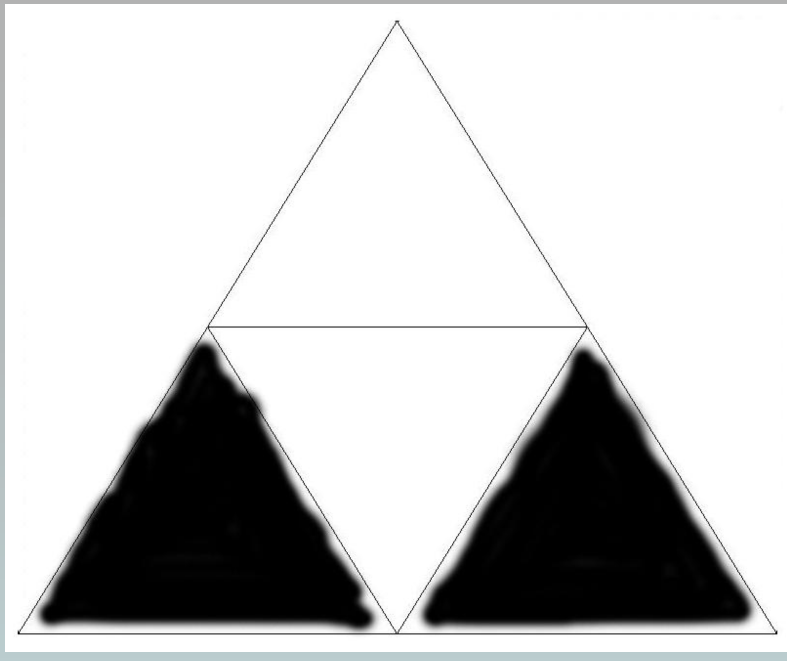
Yes: only number 3 triangles are left

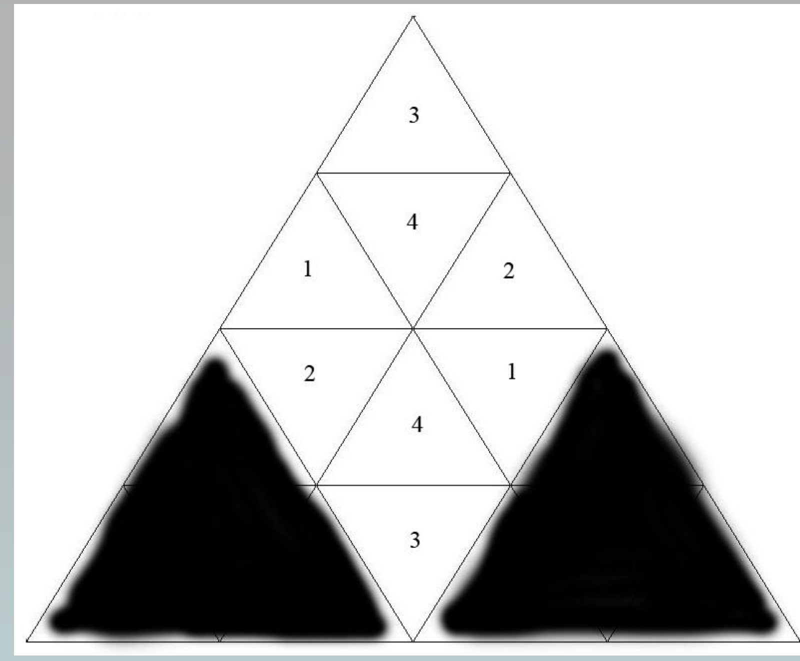


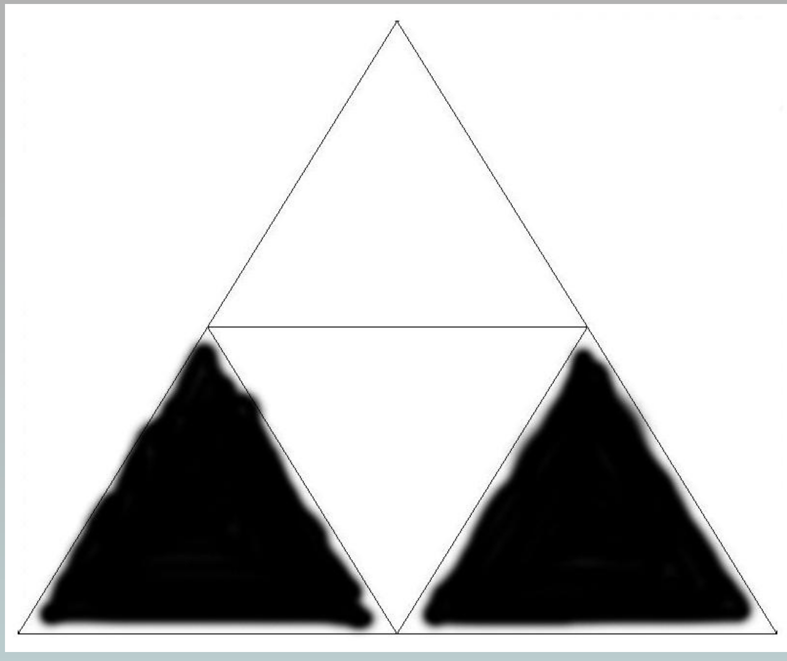
Another type: Remove triangles 1 and 2

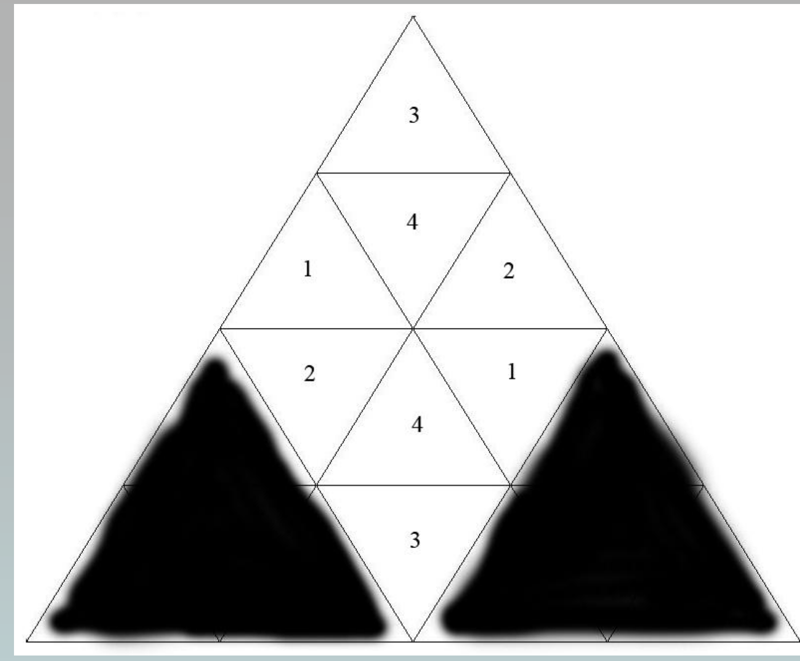


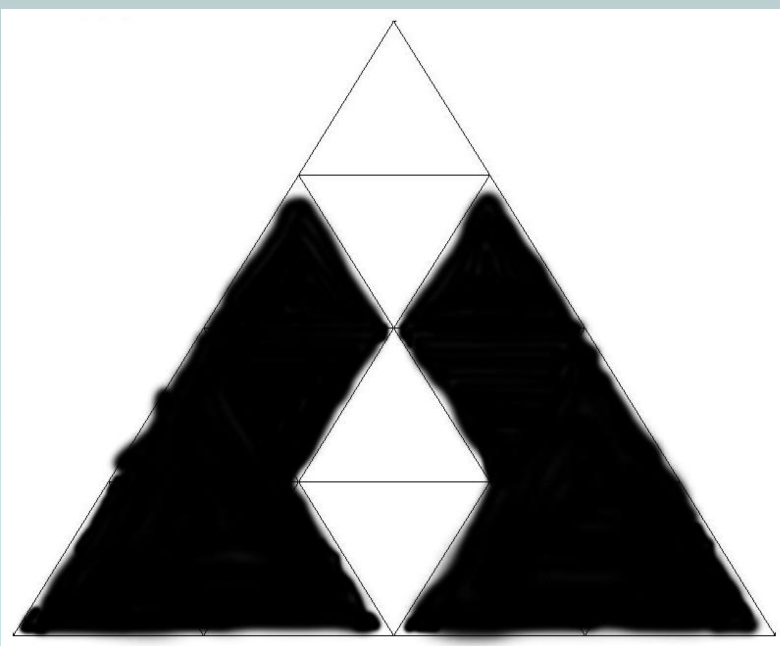


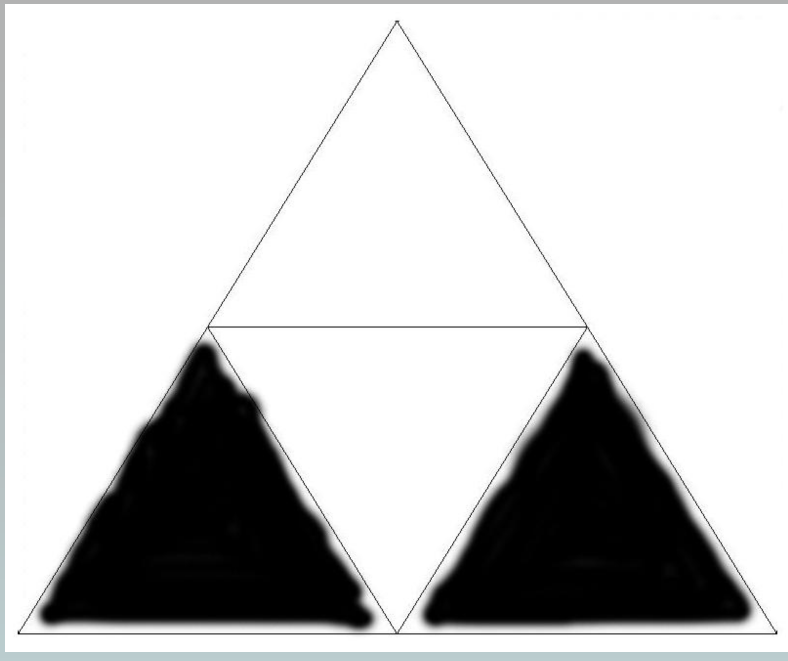


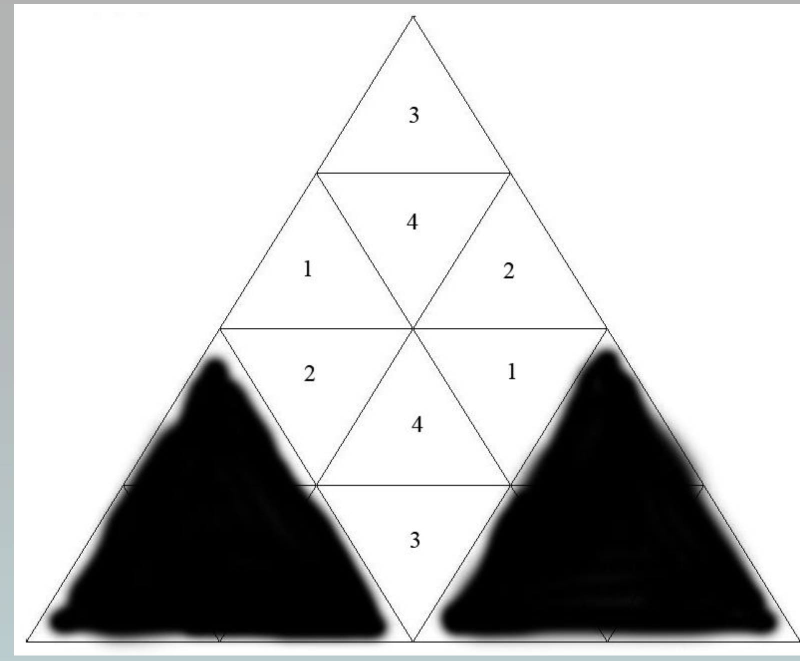

3
4
1
2
2
1
4
3

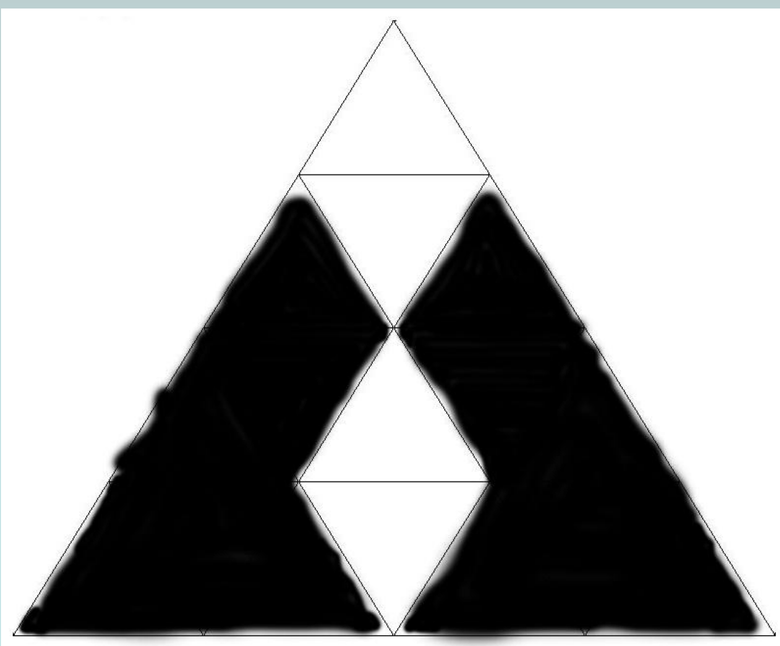


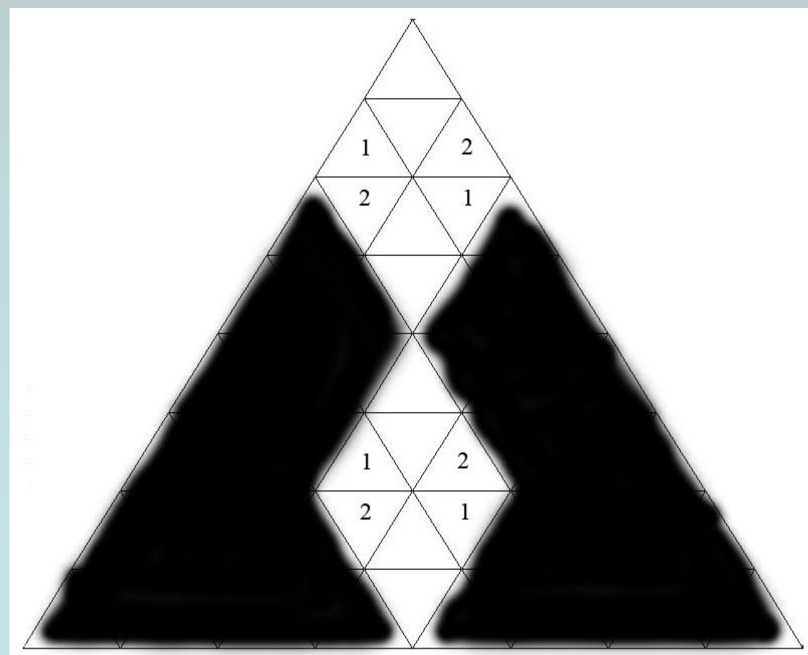

3
4
1
2
2
1
4
3

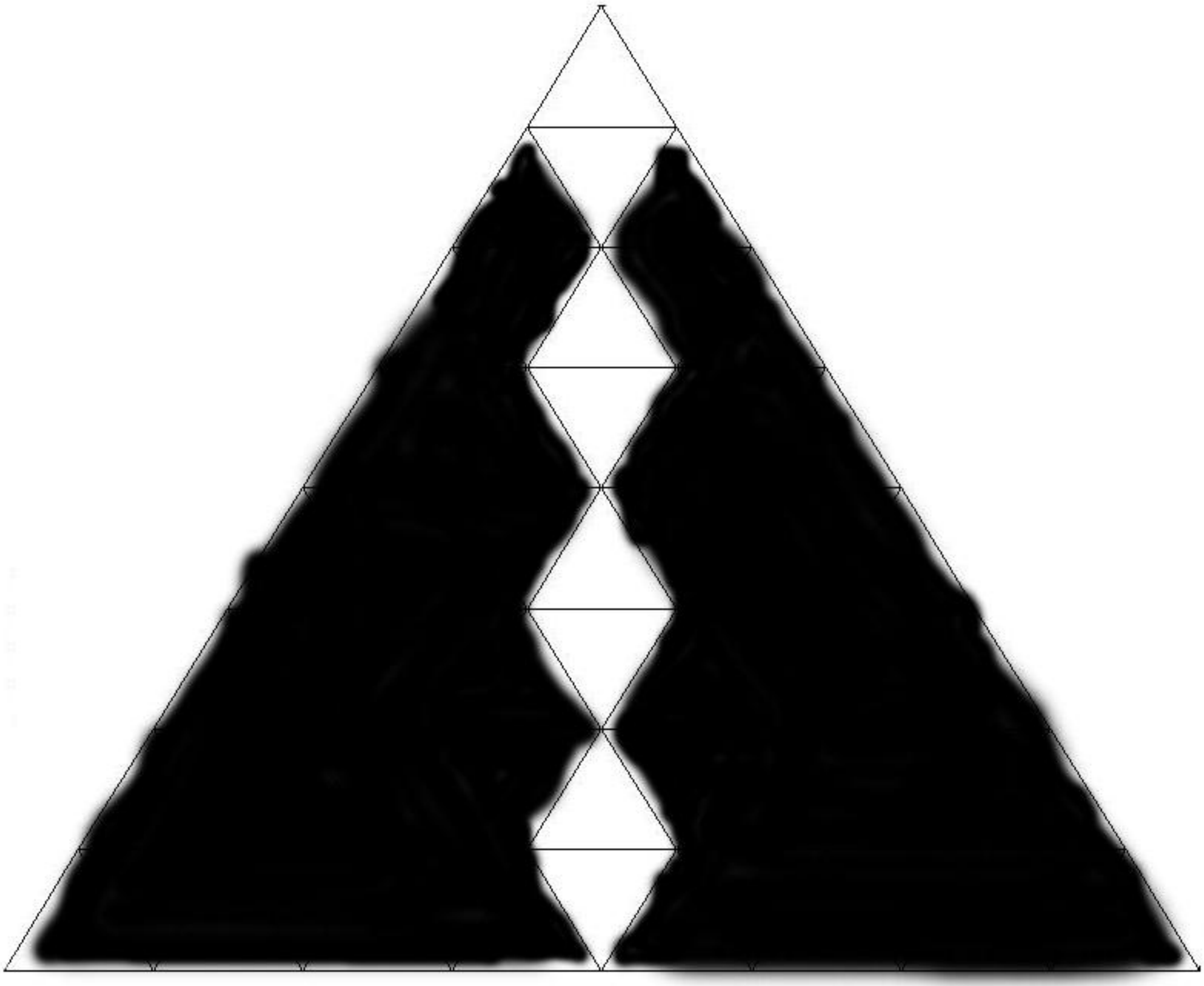




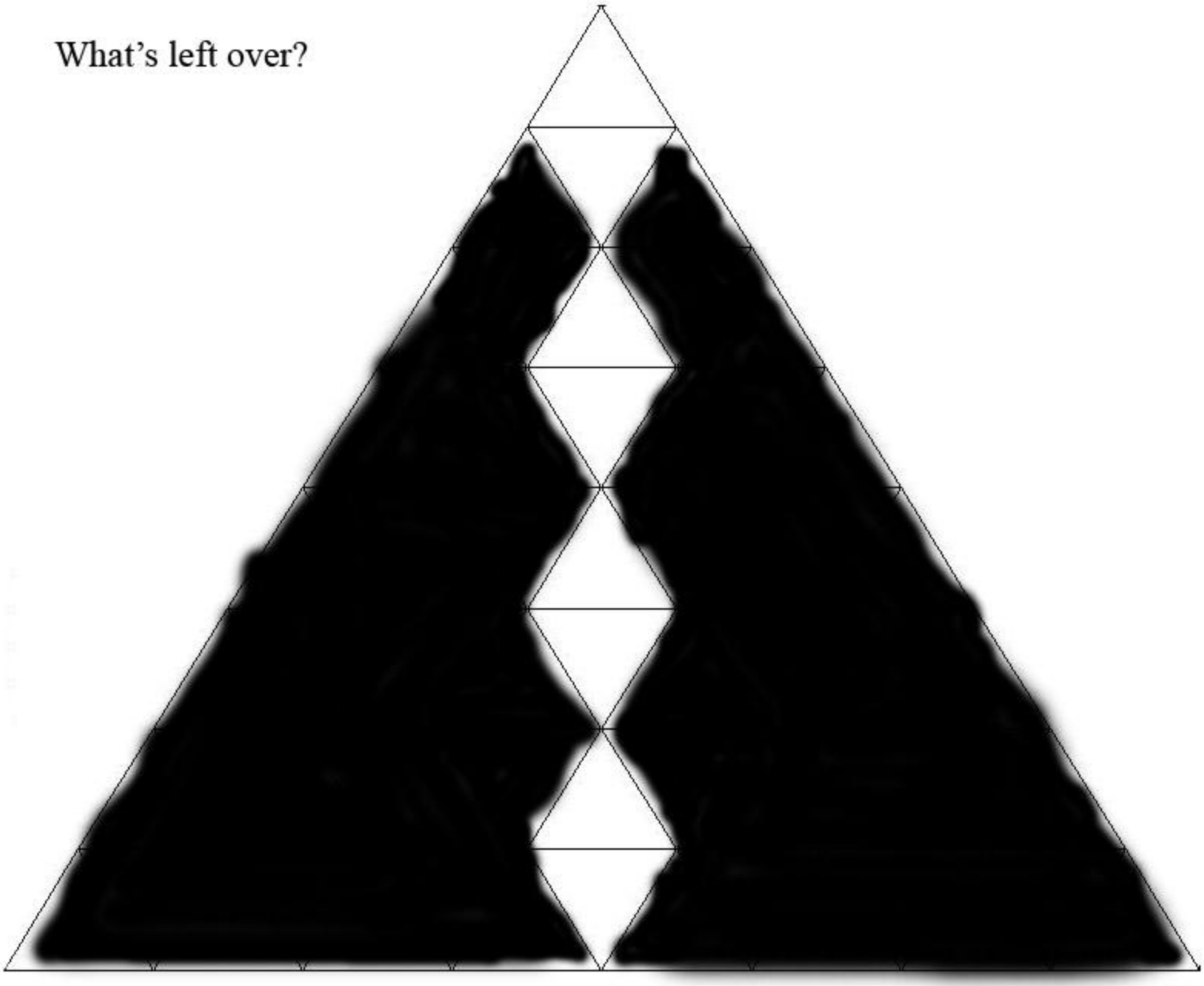

3
4
1
2
2
1
4
3



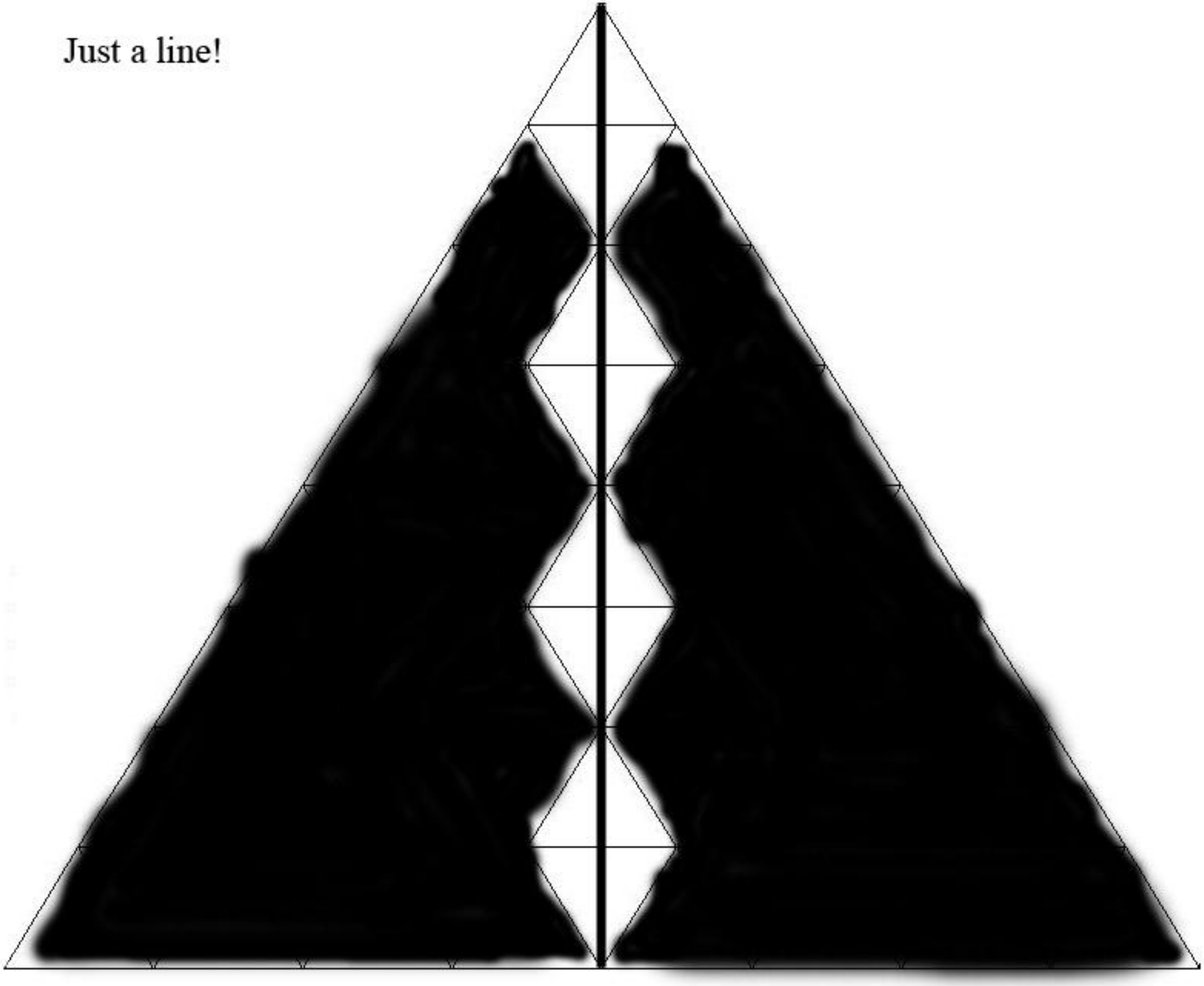

1
2
2
1
1
2
2
1



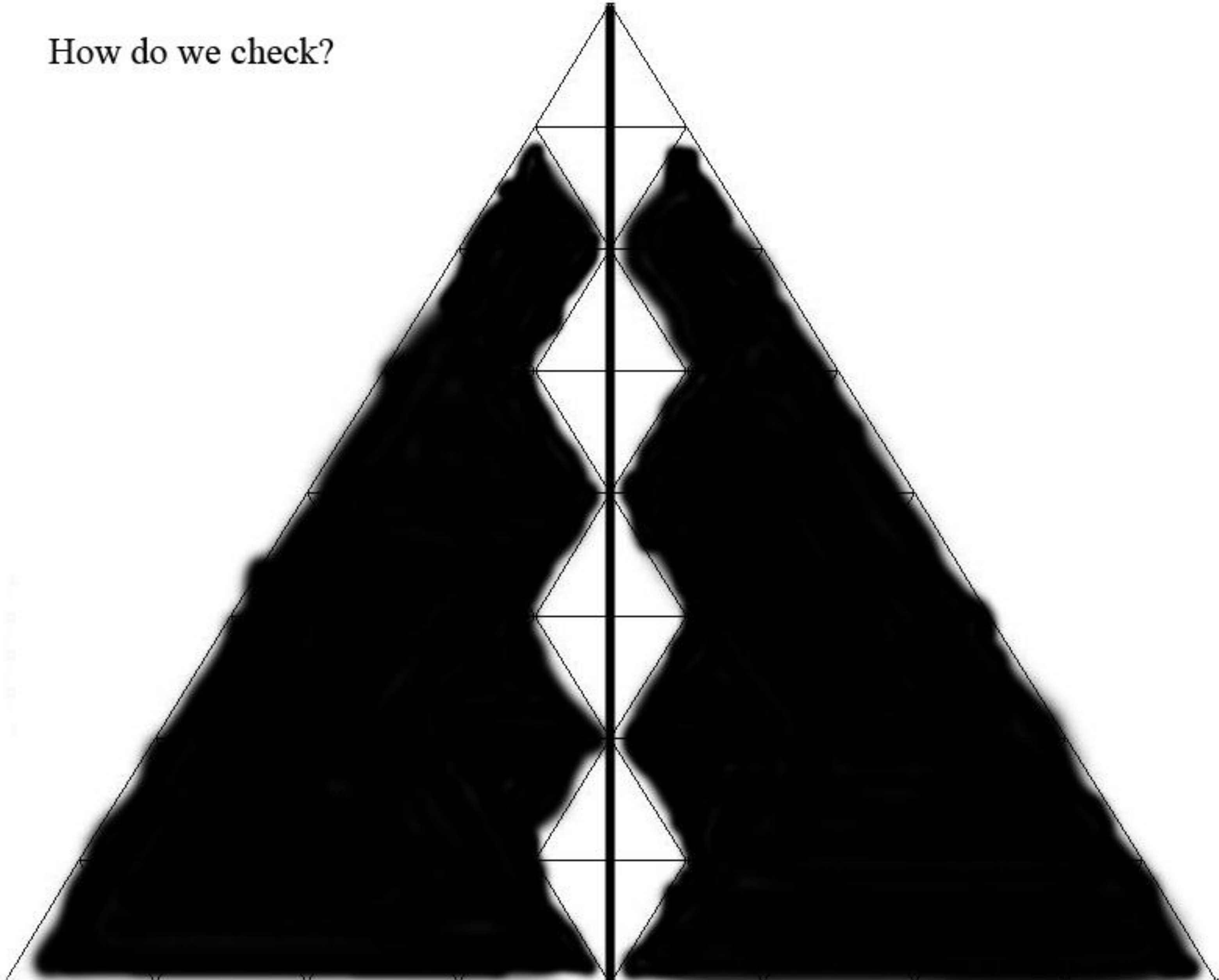
What's left over?



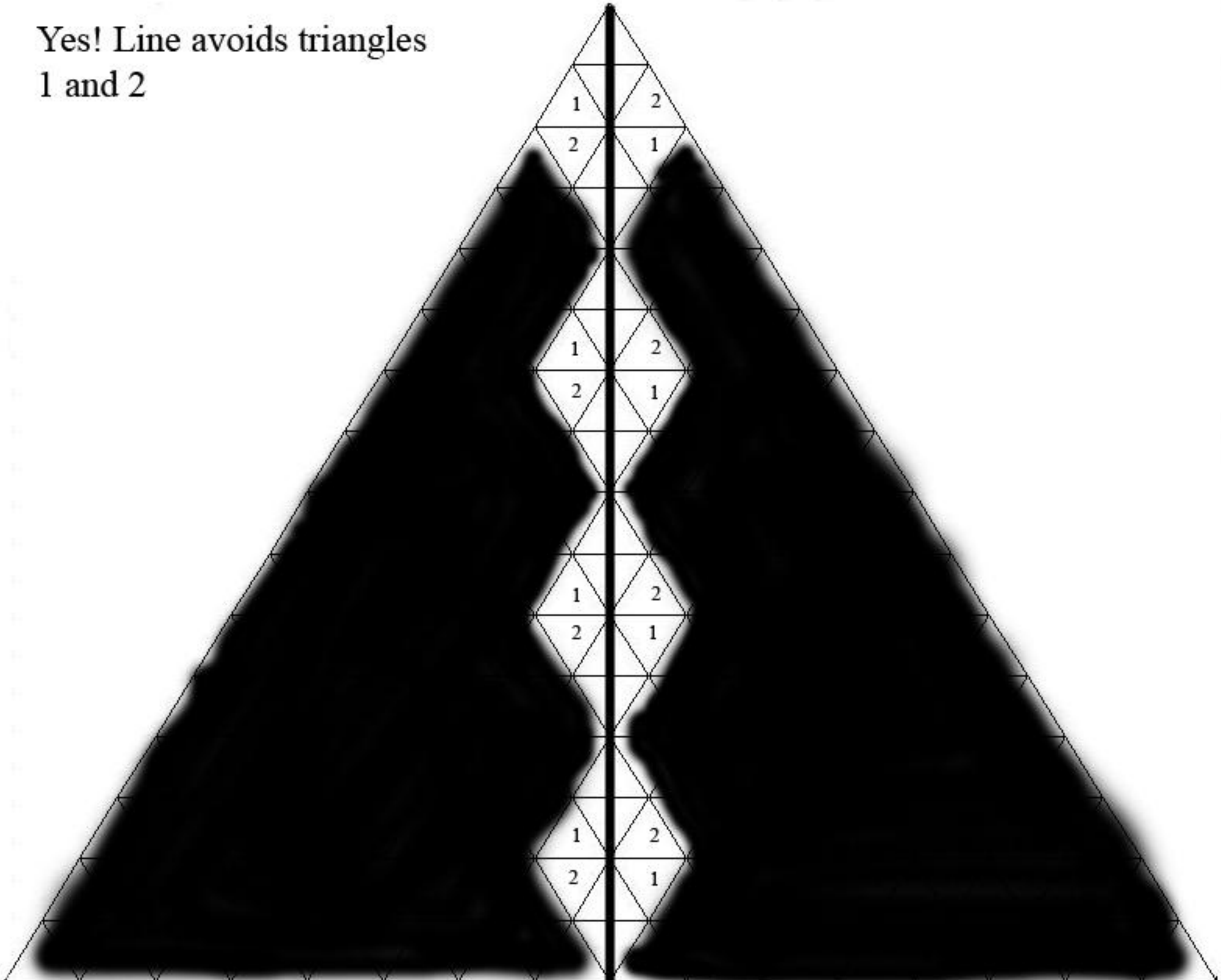
Just a line!



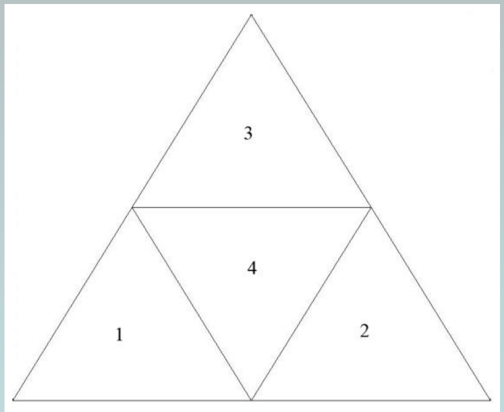
How do we check?



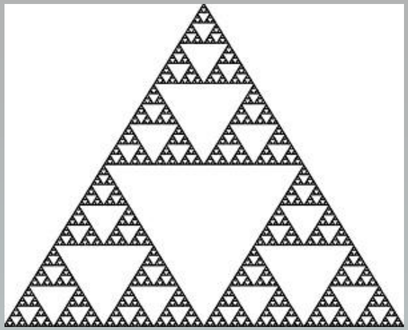
Yes! Line avoids triangles
1 and 2



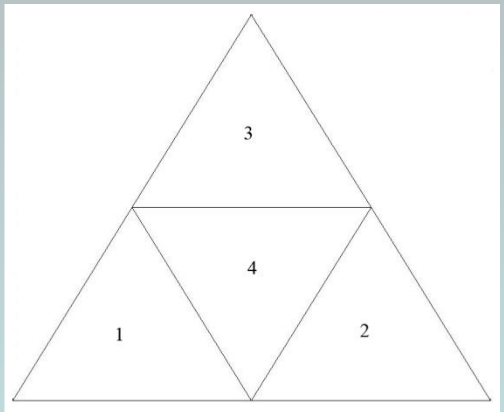
Shapes produced with the 4 triangle partition;



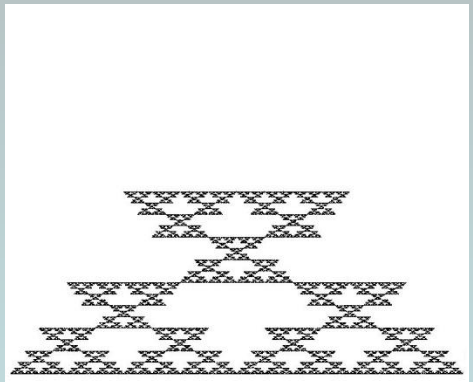
Sierpinski triangle (remove centre triangle):



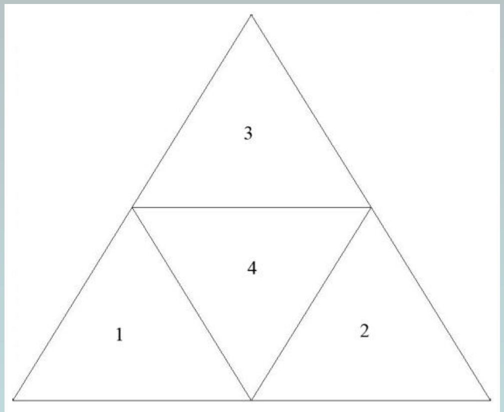
Shapes produced with the 4 triangle partition;



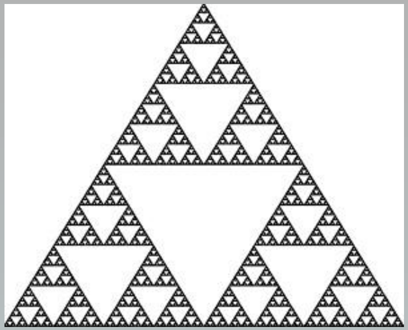
Sierpinski variation (remove corner triangle):



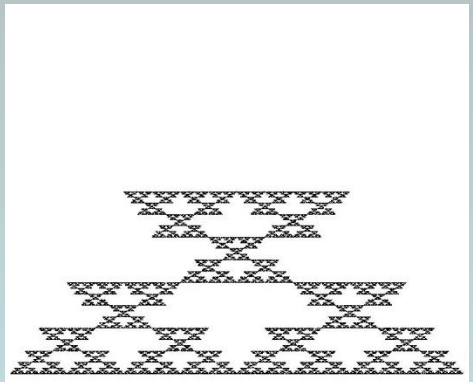
Shapes produced with the 4 triangle partition;

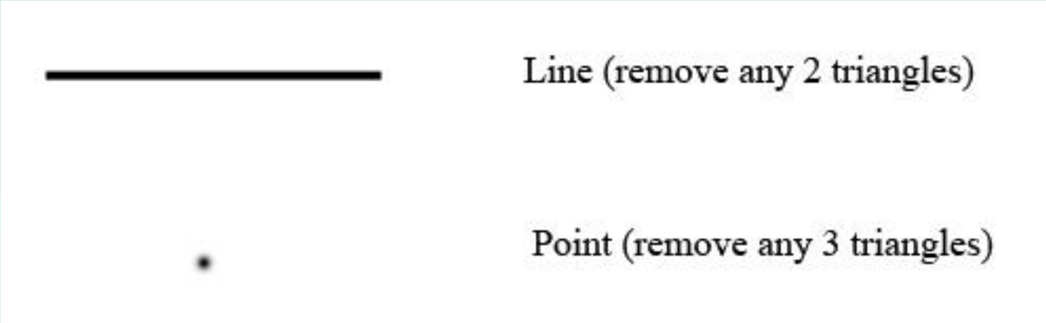


Sierpinski triangle (remove centre triangle):

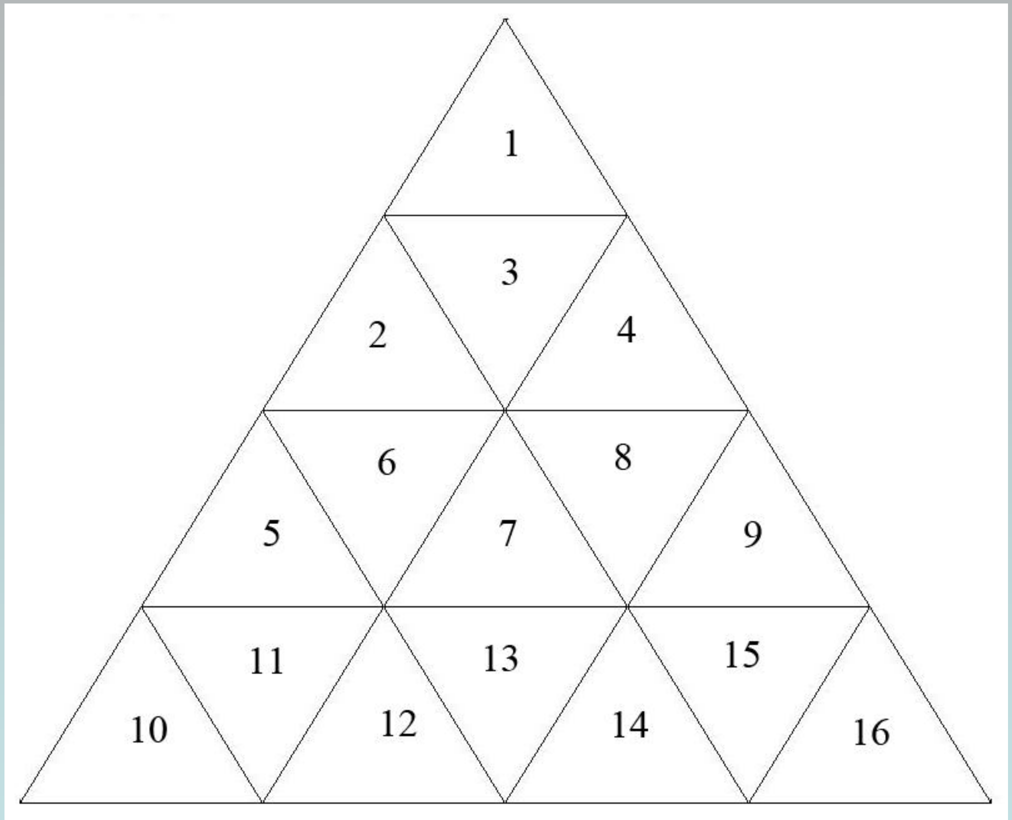


Sierpinski variation (remove corner triangle):





For more variety, begin with the partition of the triangle into 16 smaller triangles;

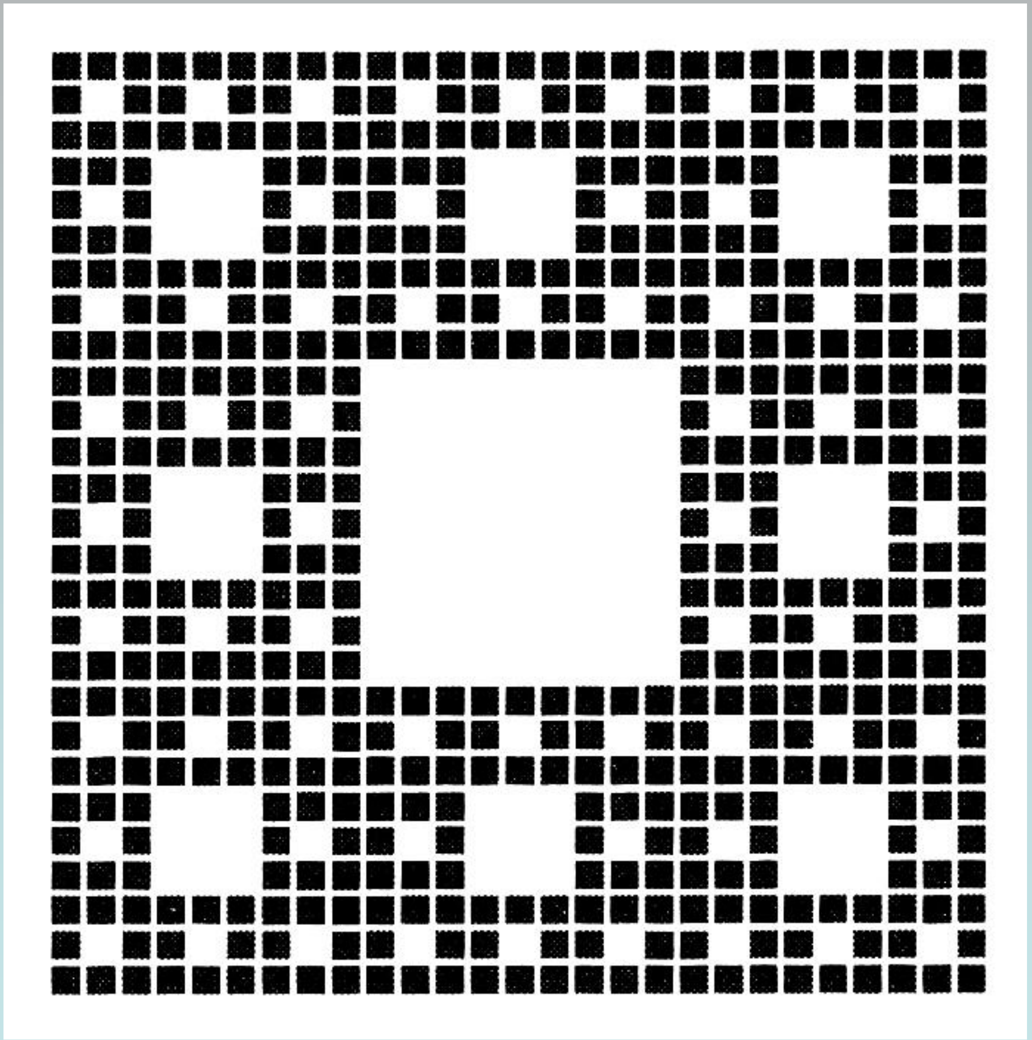


How many different shapes can you make?

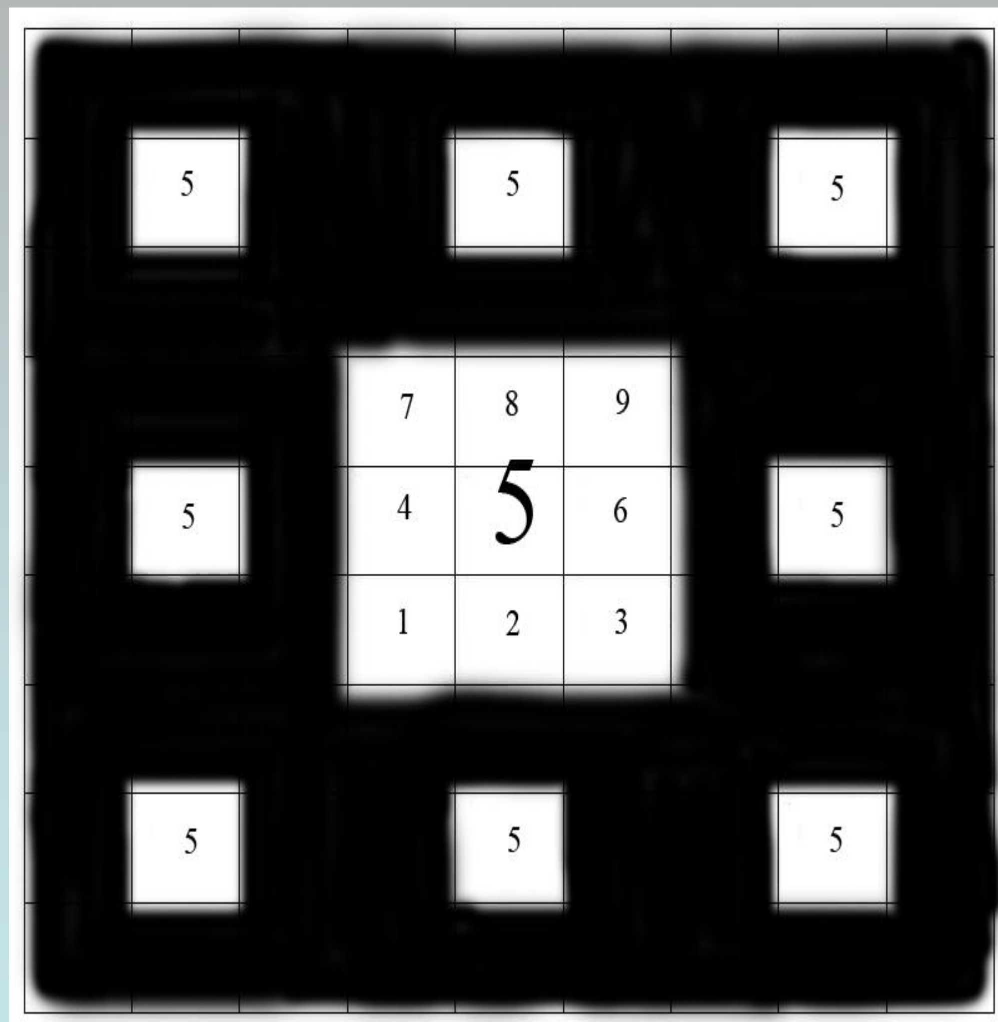
Can do the same with a square

7	8	9
4	5	6
1	2	3

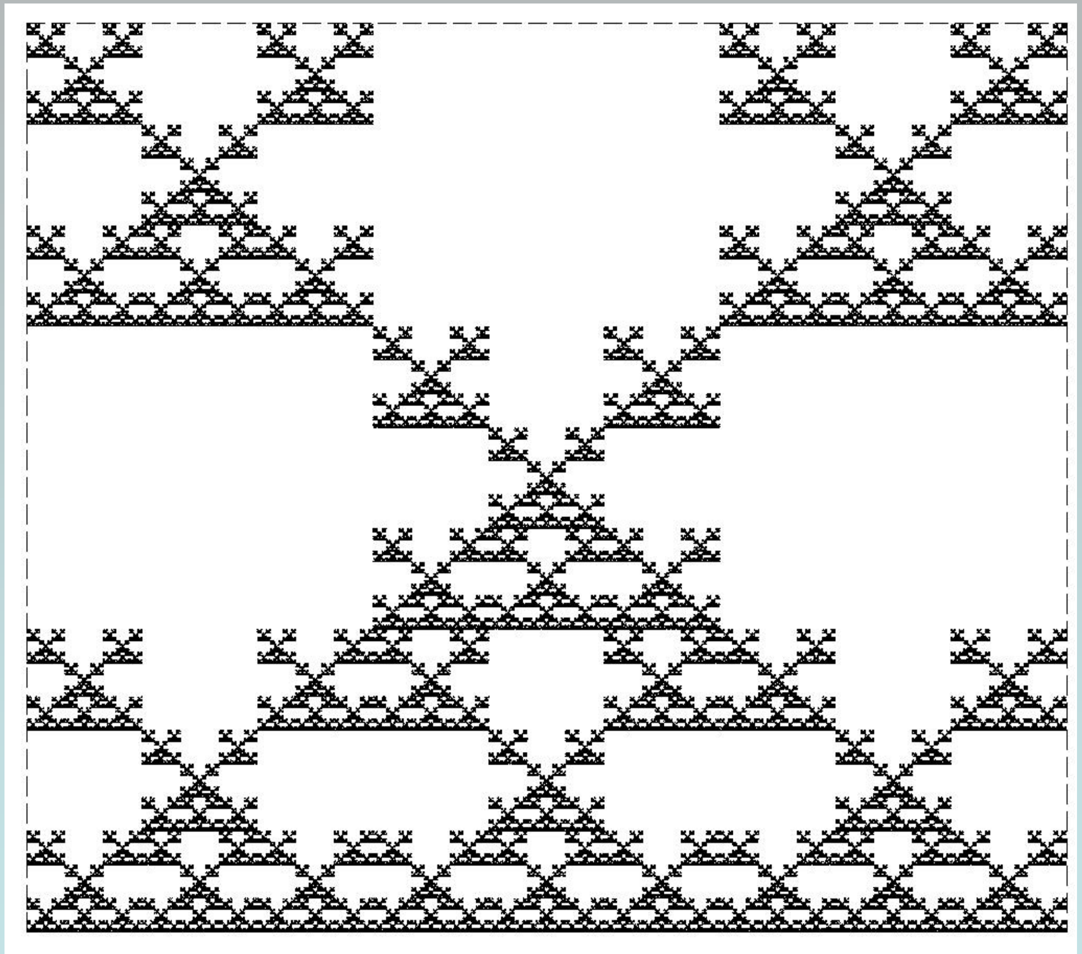
What is the ‘recipe’ for this one?



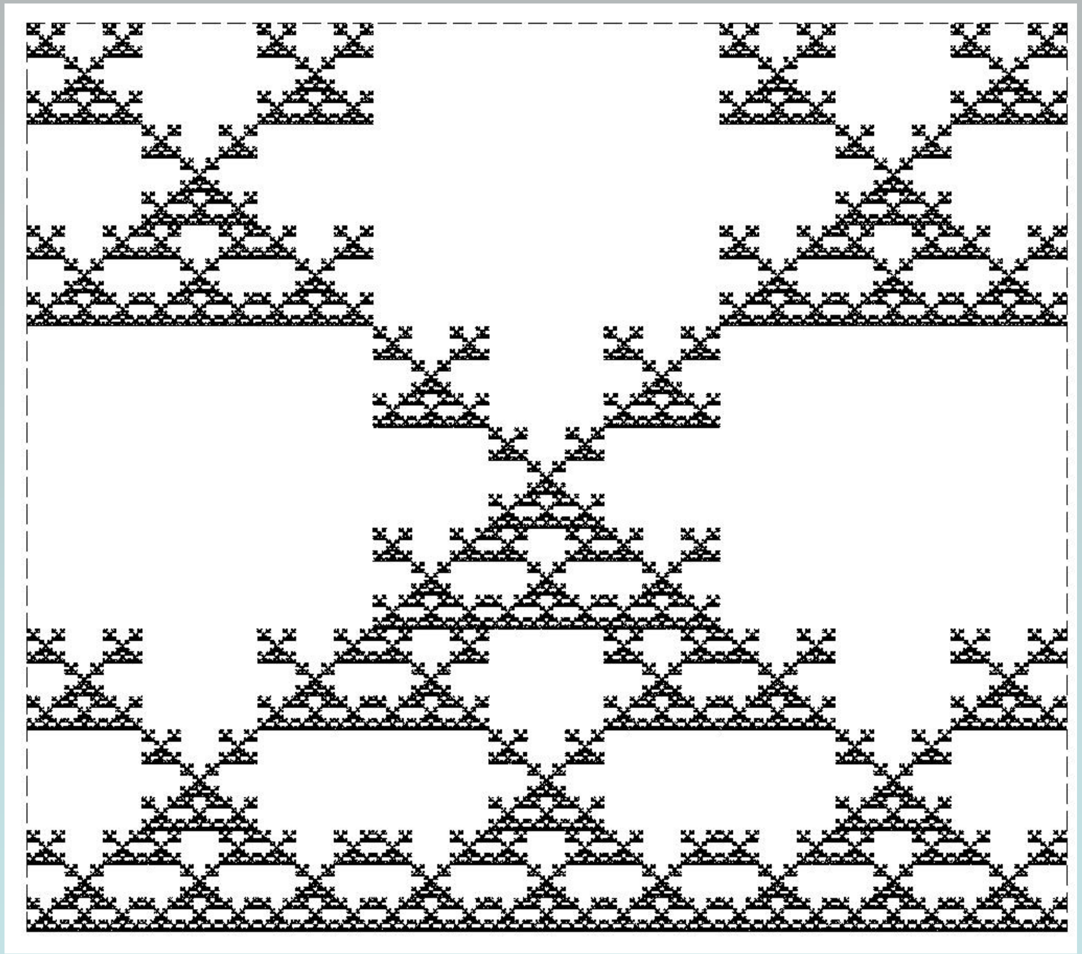
“Remove all number 5 squares”



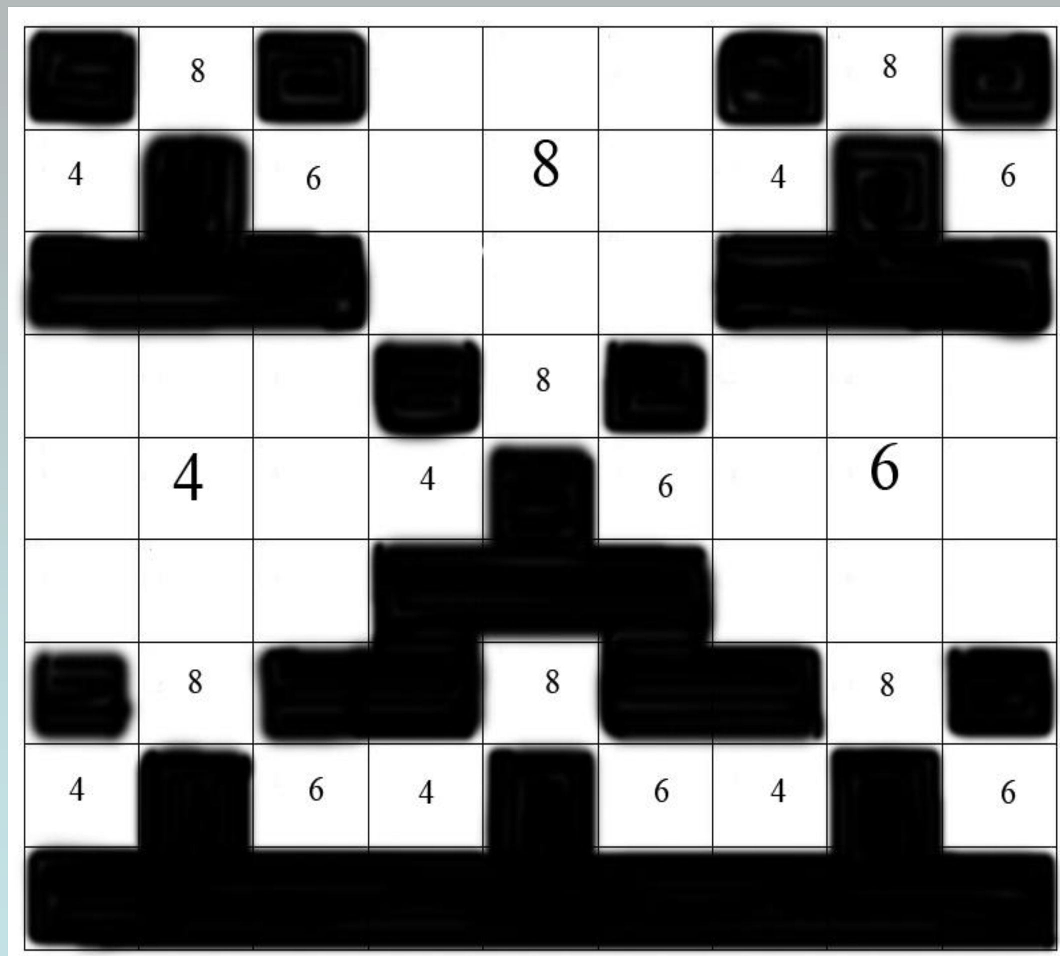
What is the ‘recipe’ for this image?



7	8	9
4	5	6
1	2	3



Remove all 4, 6, and 8 squares

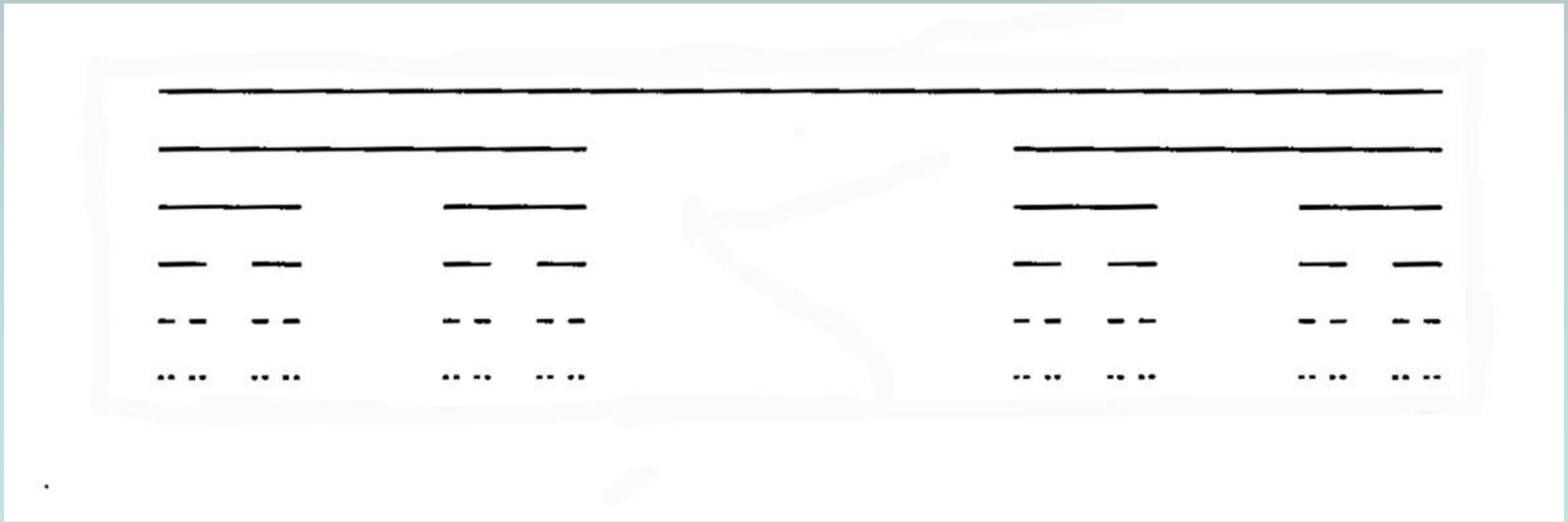


The Cantor Set

Begin with the closed interval [0, 1].
Remove the open middle third; (1/3, 2/3). Left with [0,1/3] ∪ [2/3, 1].
Now remove the middle third of each of these intervals. Left with

[0,1/9] ∪ [2/9,1/3] ∪ [2/3,7/9] ∪ [8/9,1]

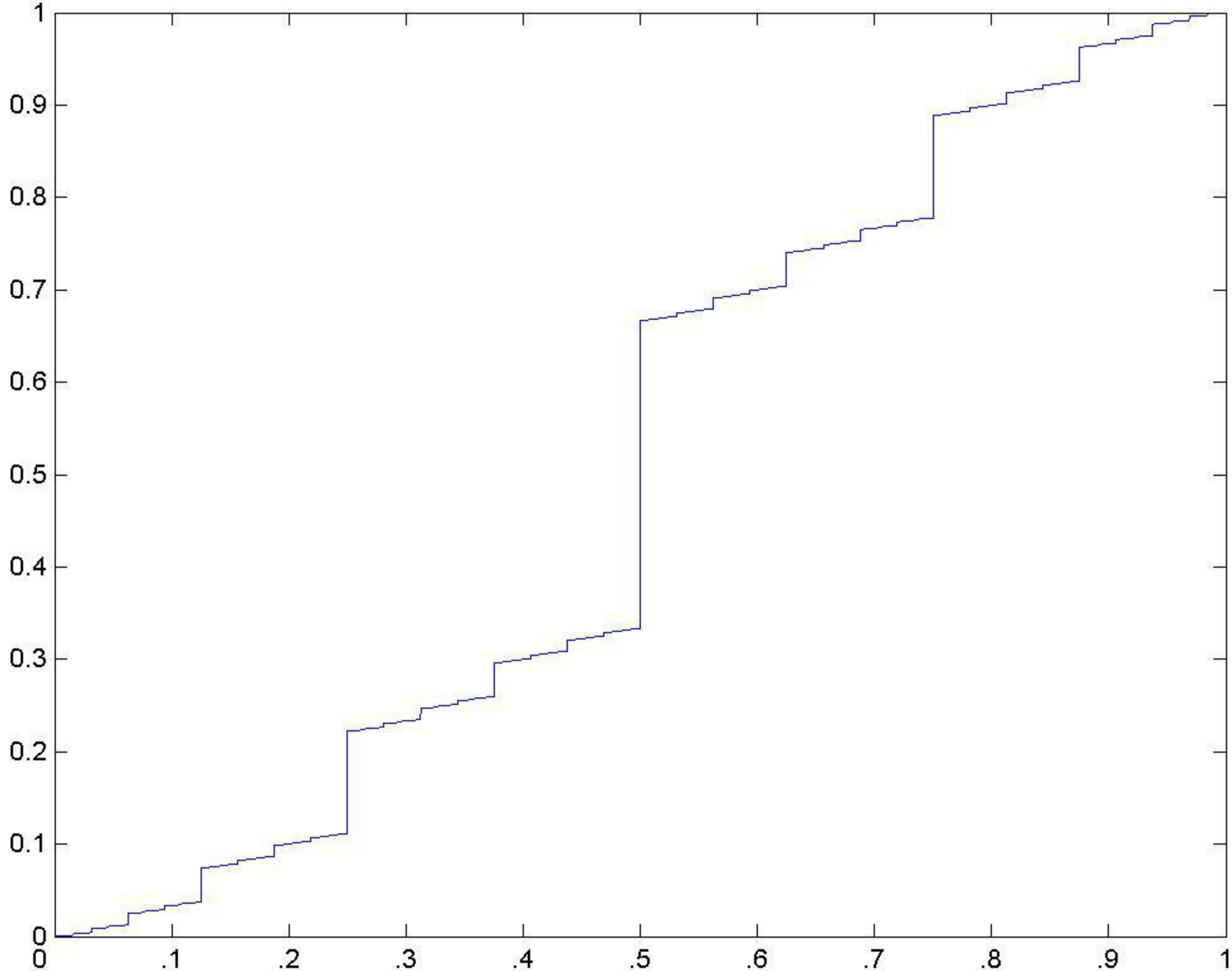
Repeat ad infinitum . . . .



Some (remarkable) properties of the Cantor set C:

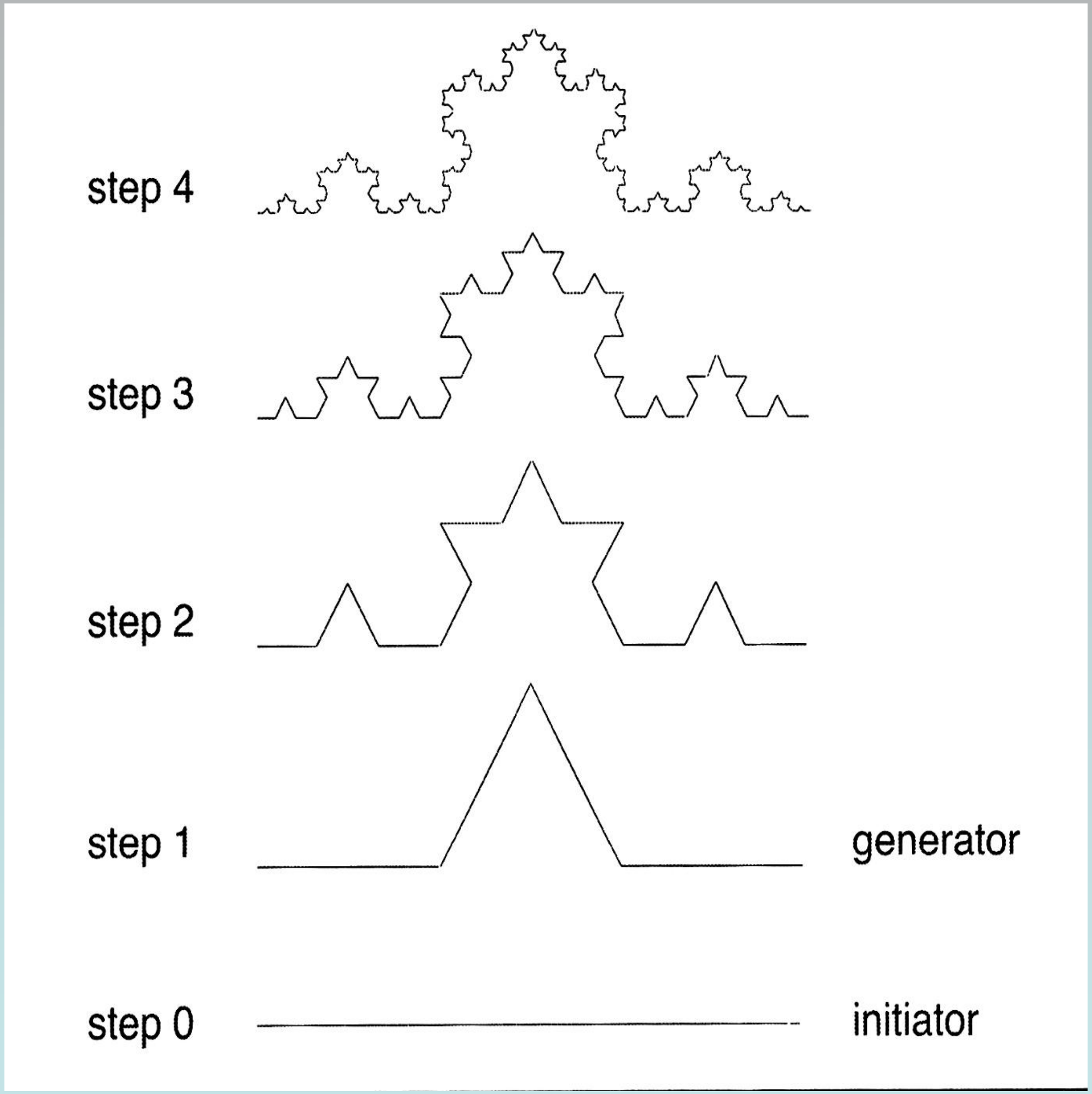
- length is zero (removed a set of length 1!)
- is dust
- is self-similar
- has the same number (cardinality) of points as [0,1]!!

In other words, we ‘rearranged’ the points in [0,1]
so that they now have length 0

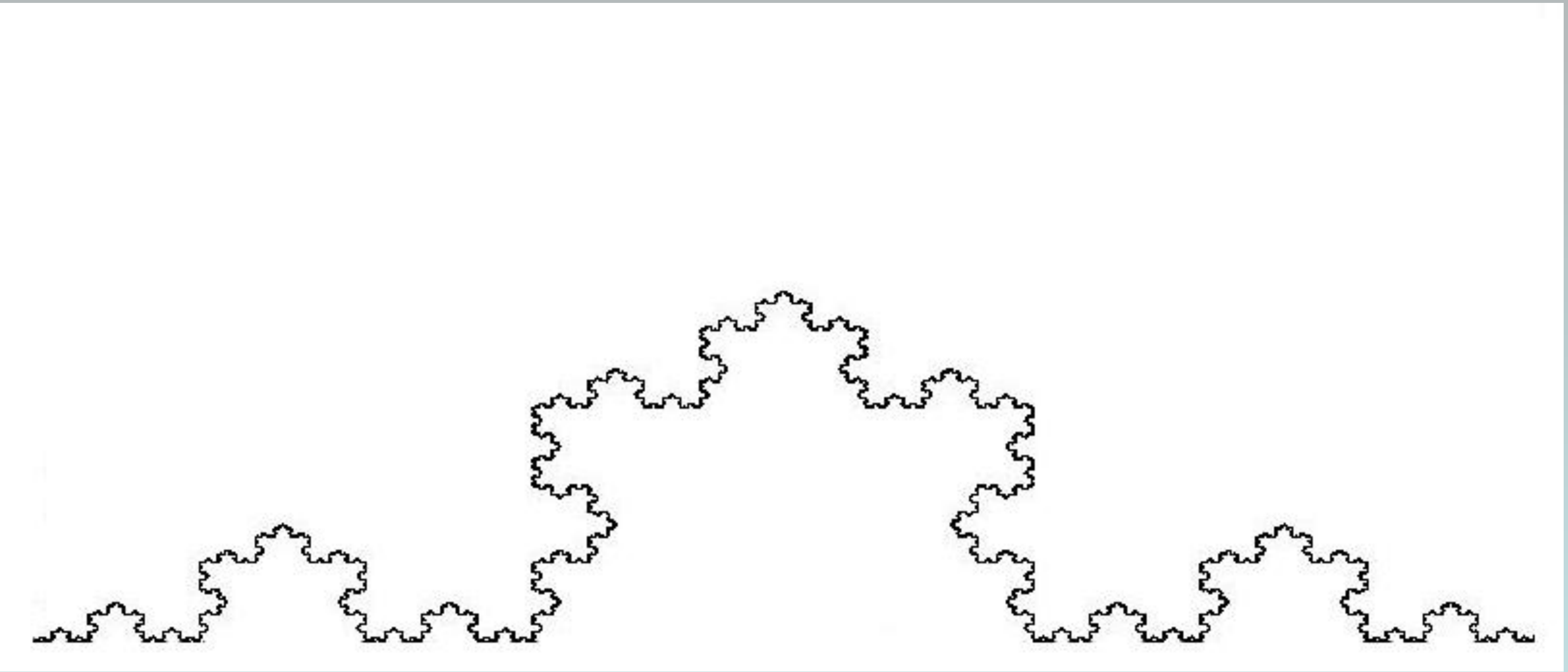


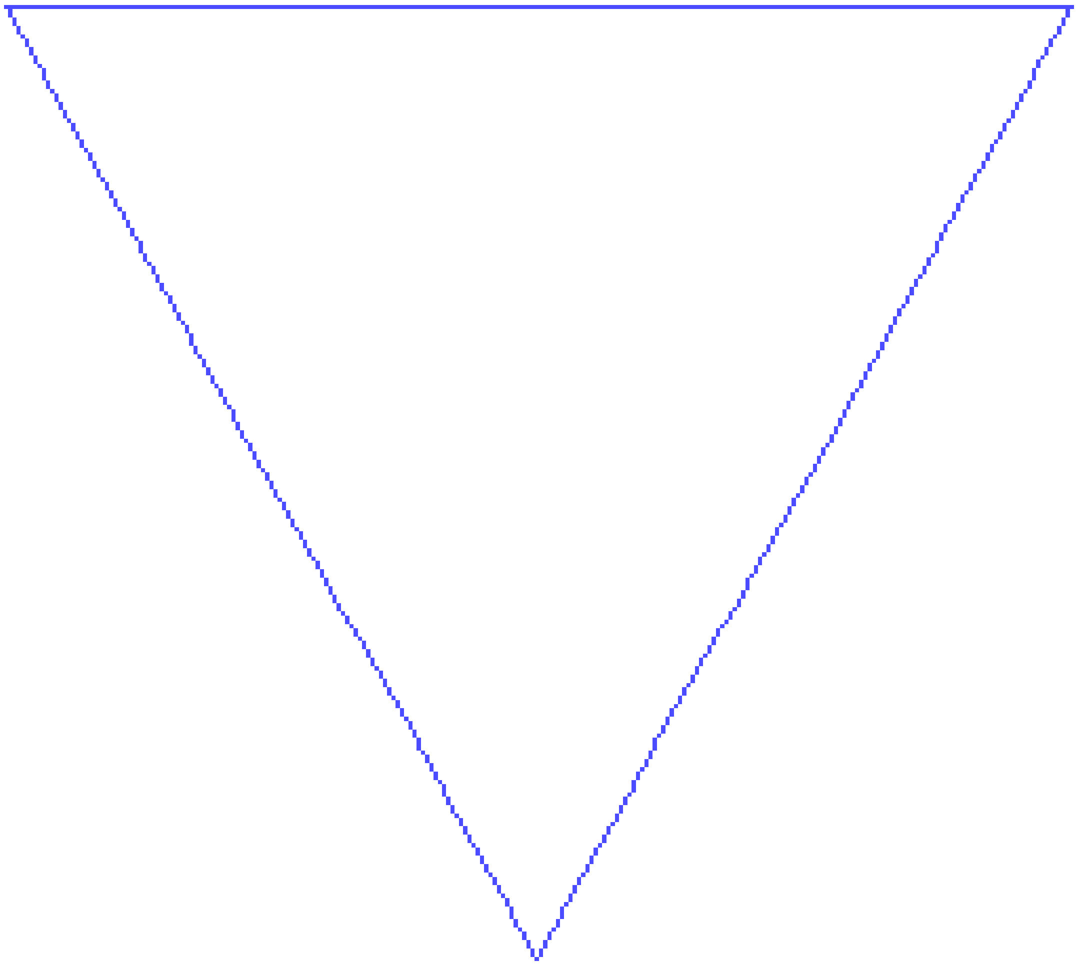
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1

Adding pieces



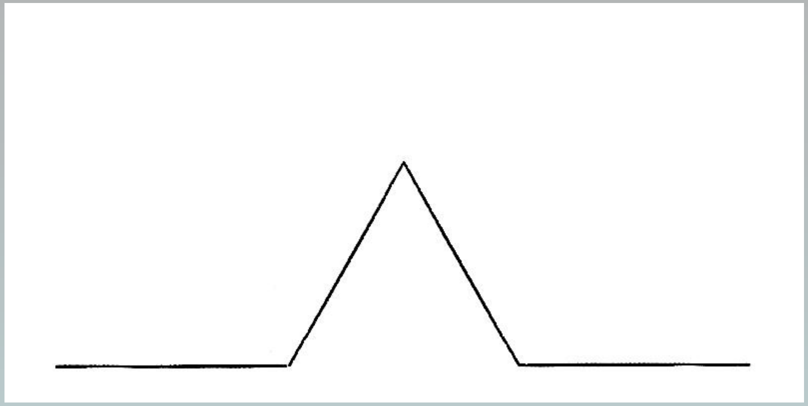
Finally…..





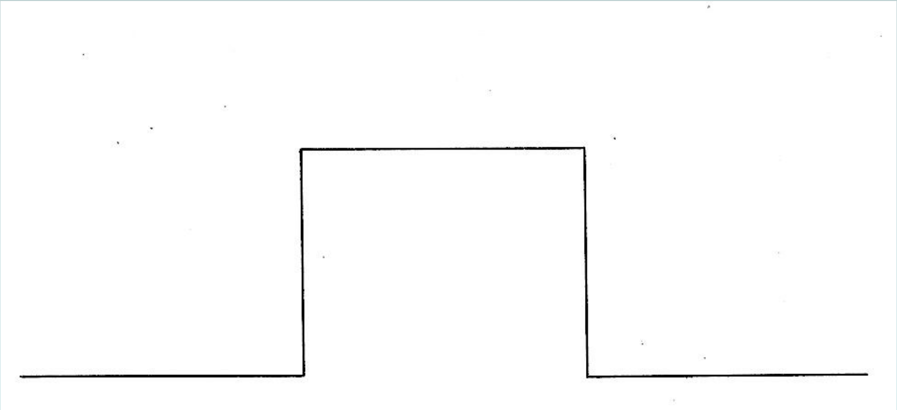
Try this:

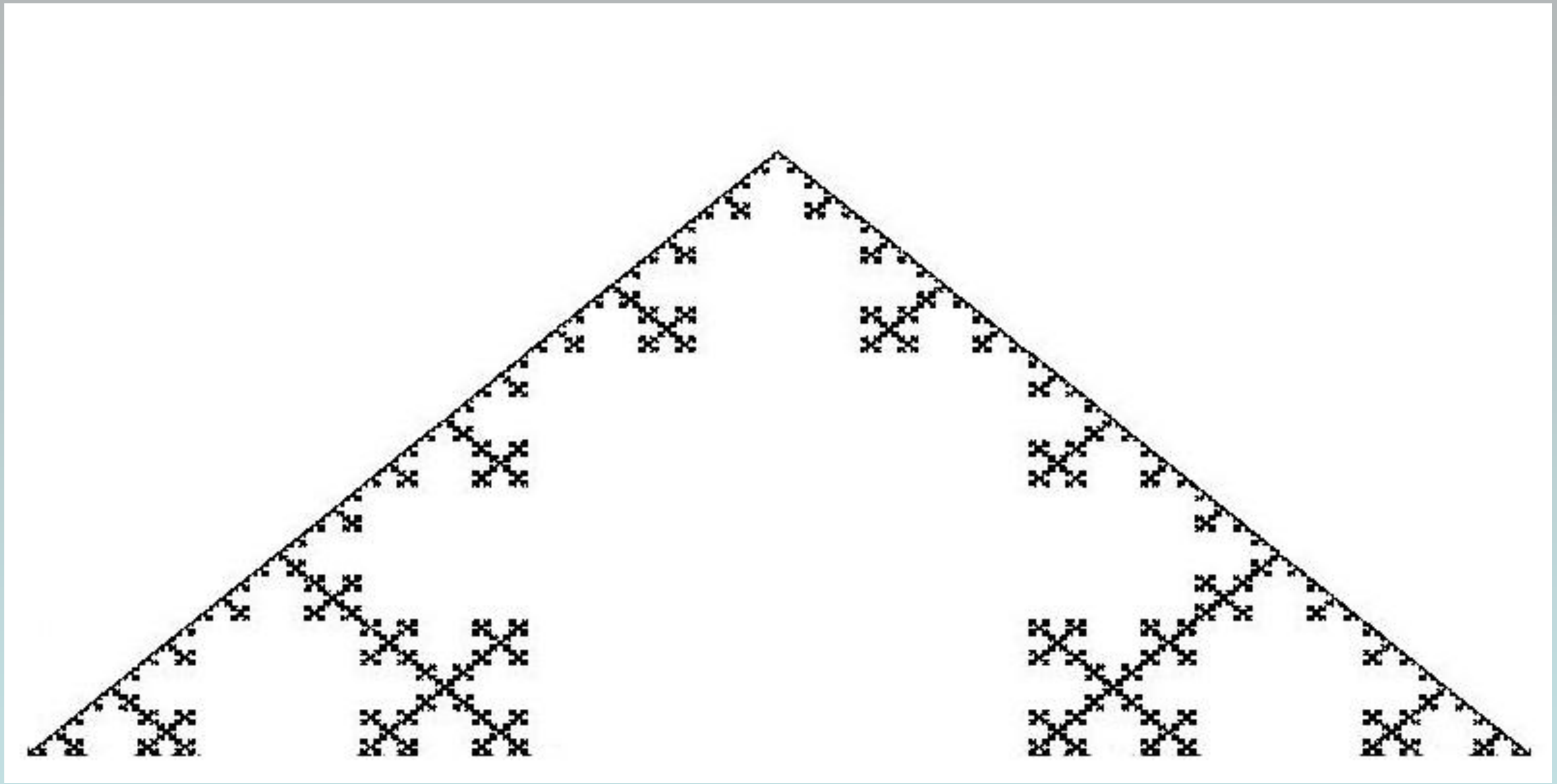
Instead of this generator



Use this one

(all sides 1/3 long)



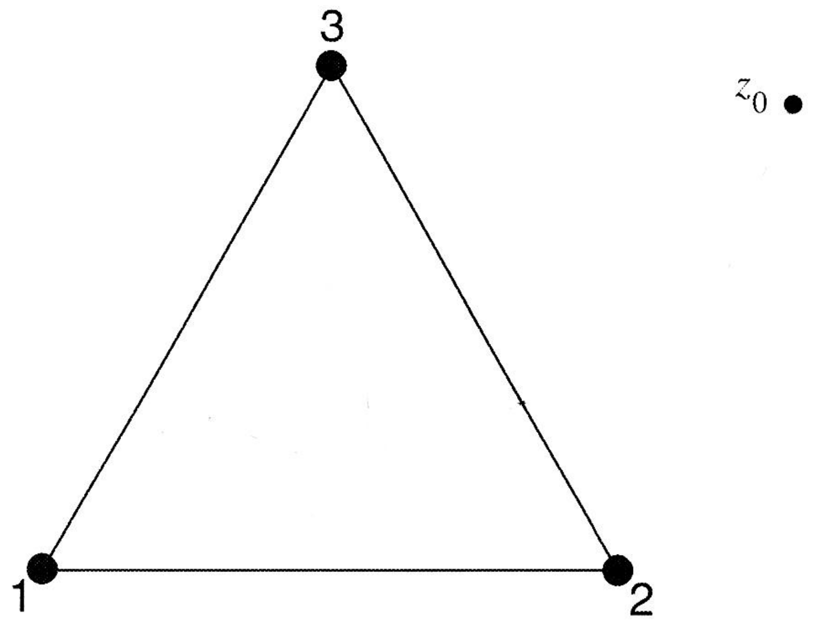


The Chaos Game

Playing the Chaos game to draw fractals

Sierpinski (Triangle)

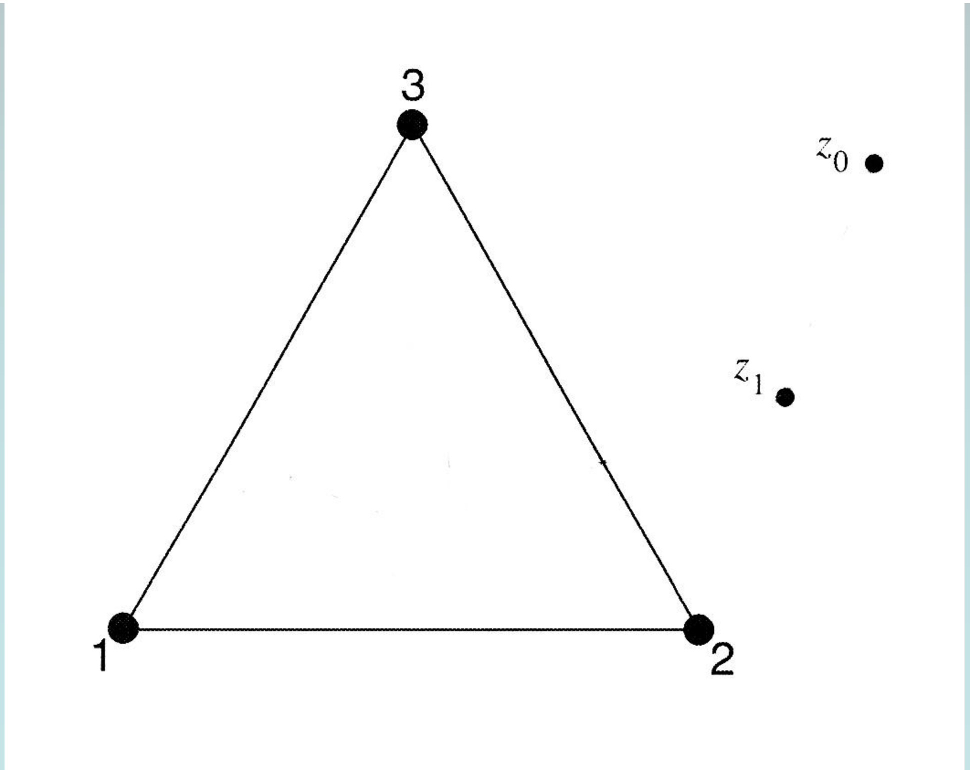
- **three pins 1, 2, 3, arranged at vertices of equilateral triangle**
- **choose random number s_i from** $\{1, 2, 3\}$
- **move 1/2 distance from current game point to black pin labelled s_i**



Playing the Chaos game to draw fractals

Sierpinski (Triangle)

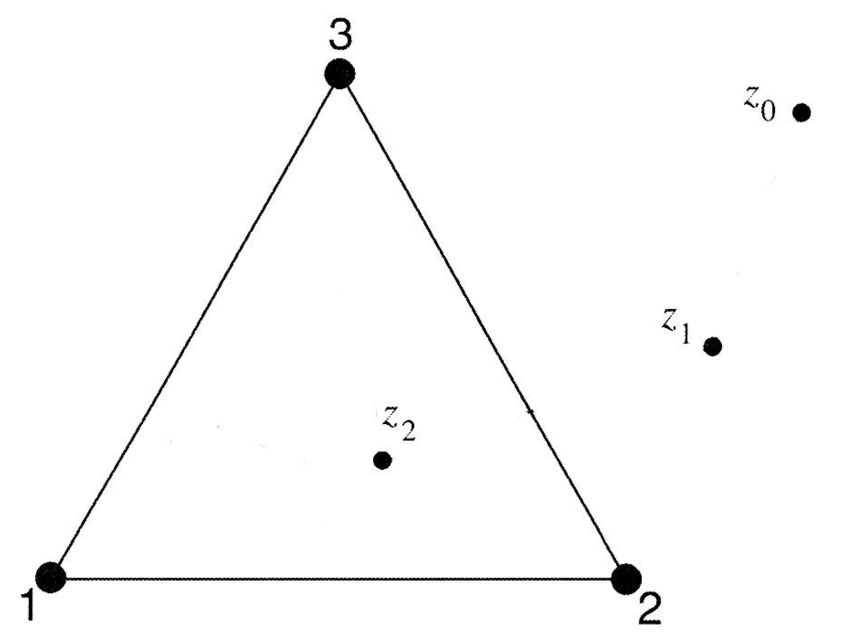
- **three pins 1, 2, 3, arranged at vertices of equilateral triangle**
- **choose random number s_i from** $\{1, 2, 3\}$
- **move 1/2 distance from current game point to black pin labelled s_i**



Playing the Chaos game to draw fractals

Sierpinski (Triangle)

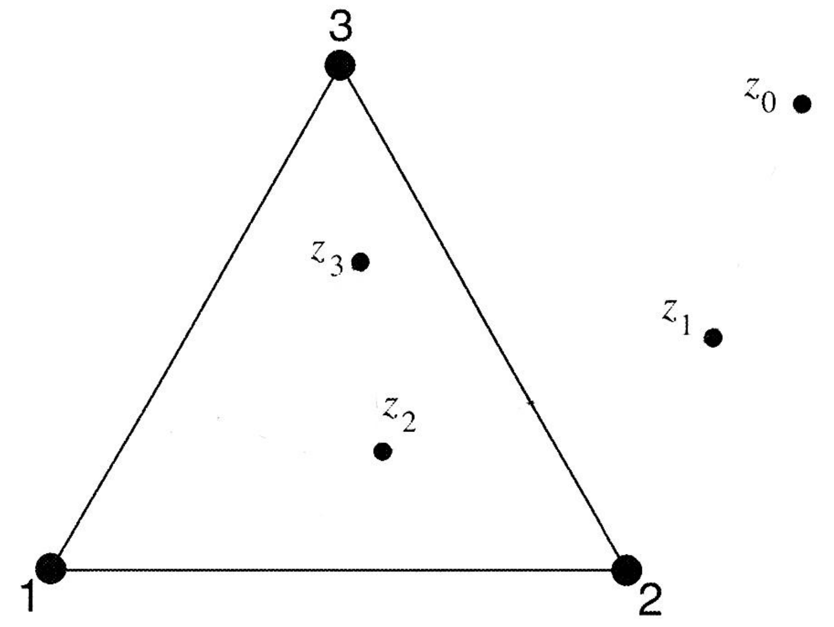
- **three pins 1, 2, 3, arranged at vertices of equilateral triangle**
- **choose random number s_i from** $\{1, 2, 3\}$
- **move 1/2 distance from current game point to black pin labelled s_i**



Playing the Chaos game to draw fractals

Sierpinski (Triangle)

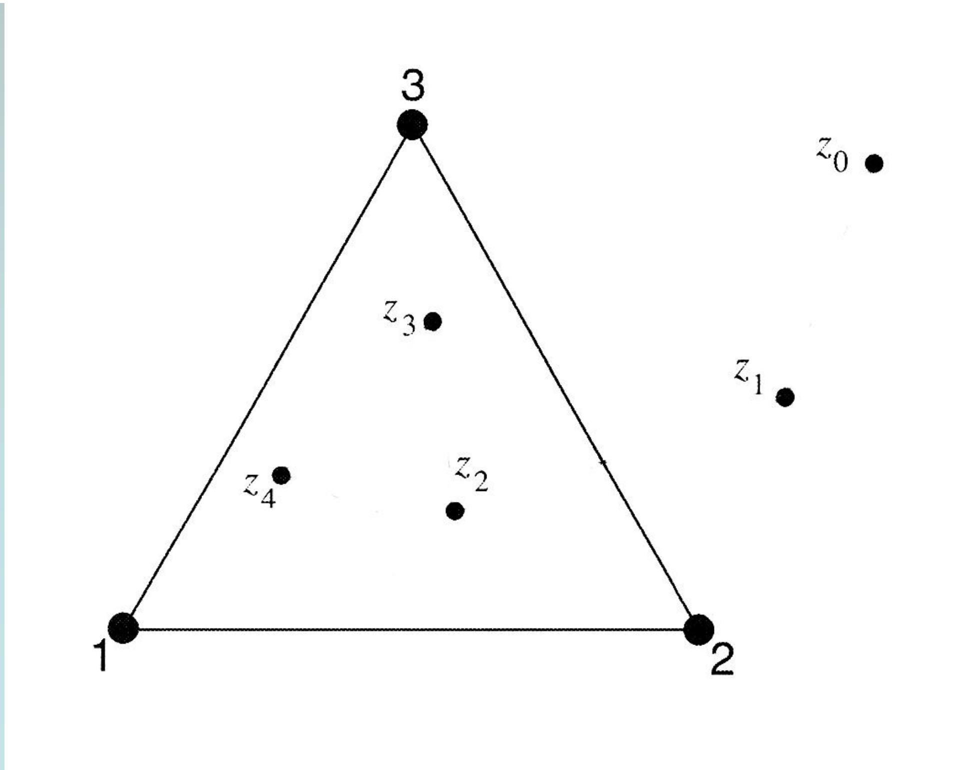
- **three pins 1, 2, 3, arranged at vertices of equilateral triangle**
- **choose random number $\mathbf{s_i}$ from** $\{1, 2, 3\}$
- **move 1/2 distance from current game point to black pin labelled $\mathbf{s_i}$**



Playing the Chaos game to draw fractals

Sierpinski (Triangle)

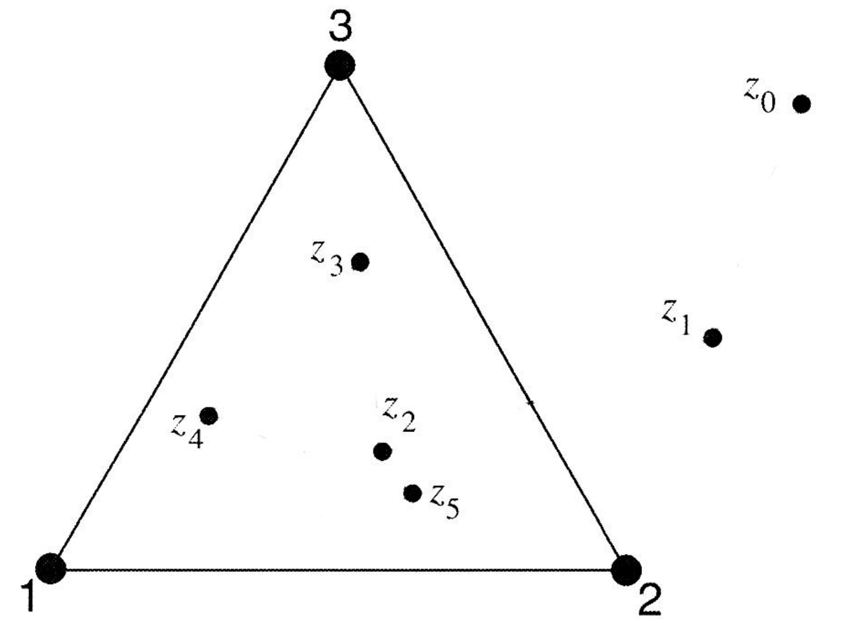
- **three pins 1, 2, 3, arranged at vertices of equilateral triangle**
- **choose random number s_i from** $\{1, 2, 3\}$
- **move 1/2 distance from current game point to black pin labelled s_i**



Playing the Chaos game to draw fractals

Sierpinski (Triangle)

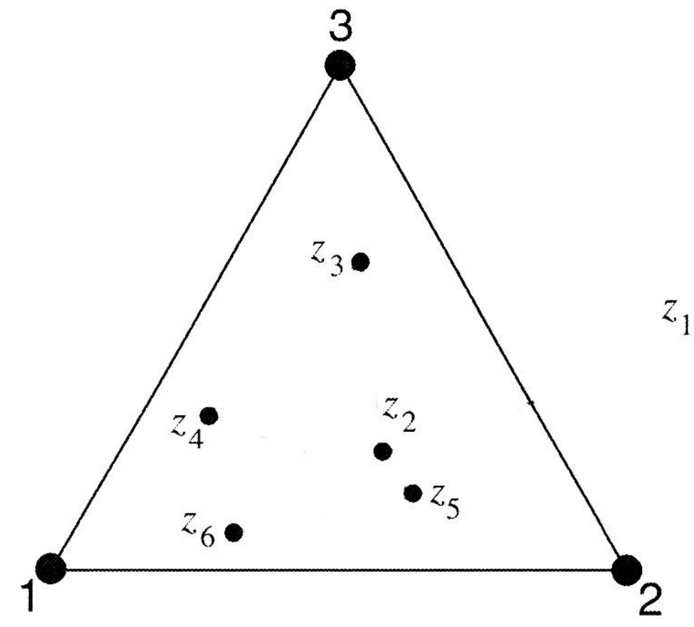
- **three pins 1, 2, 3, arranged at vertices of equilateral triangle**
- **choose random number s_i from** $\{1, 2, 3\}$
- **move 1/2 distance from current game point to black pin labelled s_i**

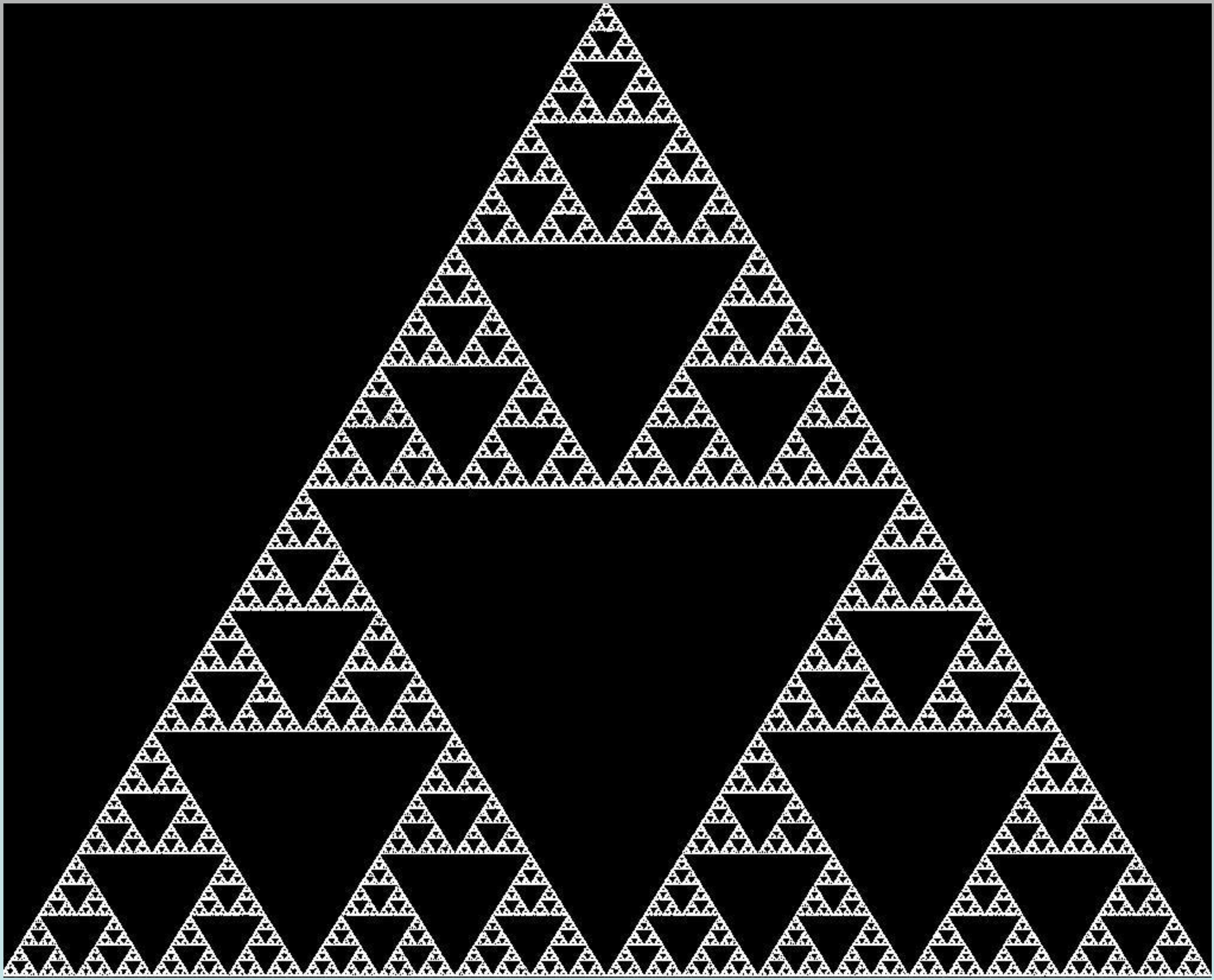


Playing the Chaos game to draw fractals

Sierpinski (Triangle)

- **three pins 1, 2, 3, arranged at vertices of equilateral triangle**
- **choose random number s_i from** $\{1, 2, 3\}$
- **move 1/2 distance from current game point to black pin labelled s_i**

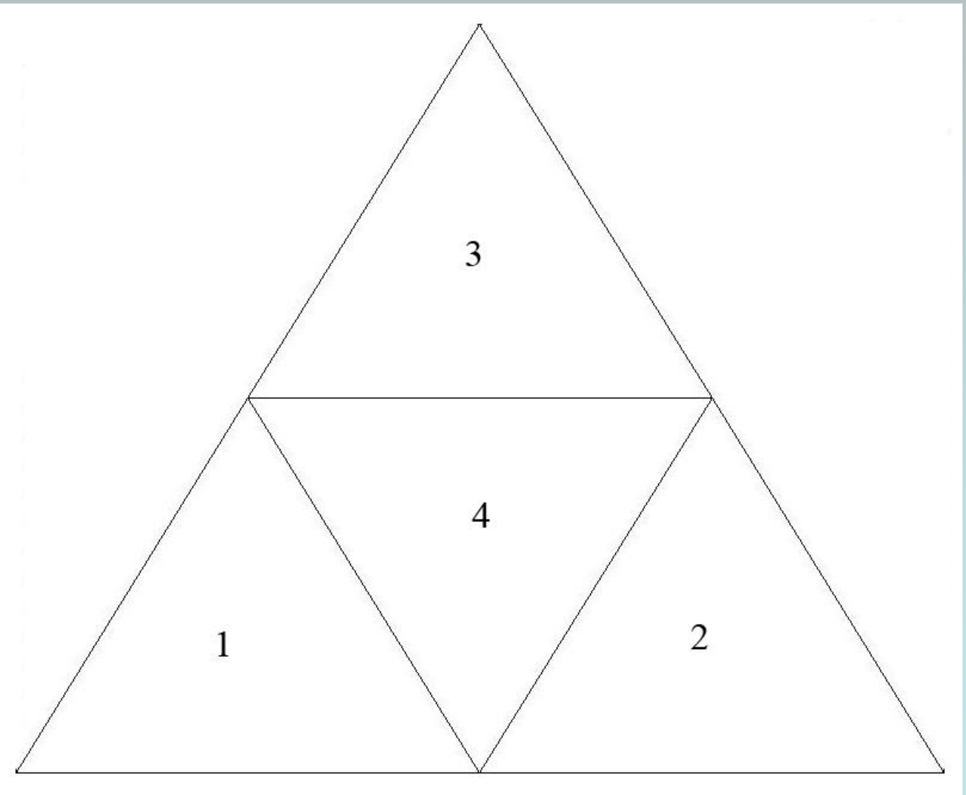




Why the Chaos Game Works

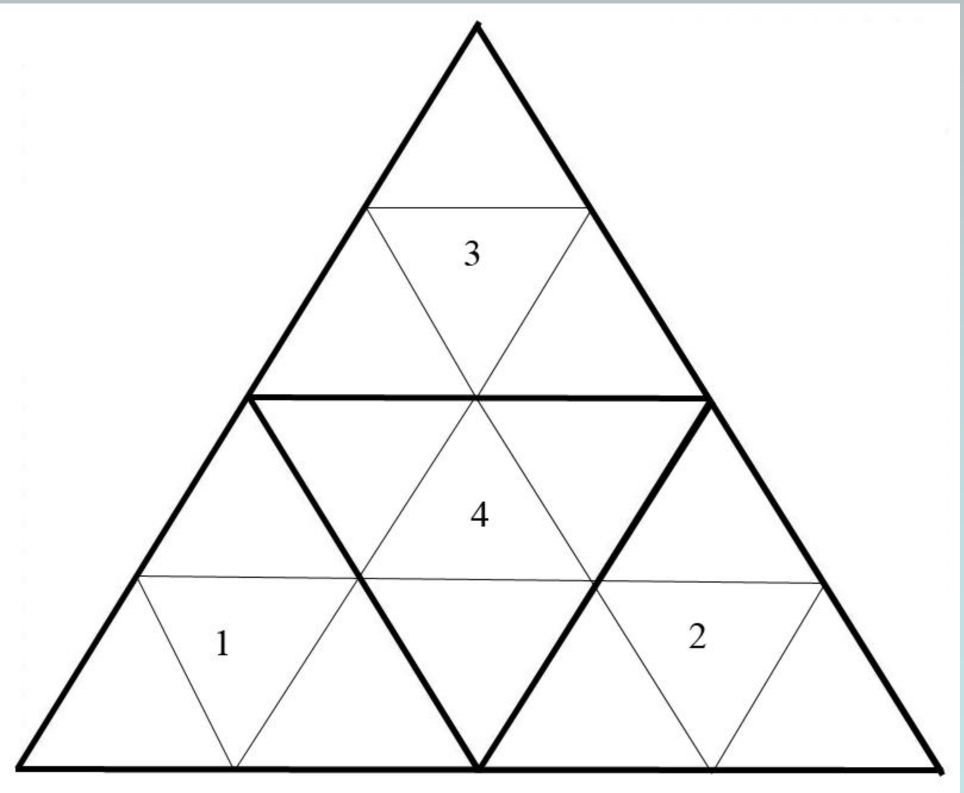
Why the Chaos Game Works

- Addresses:



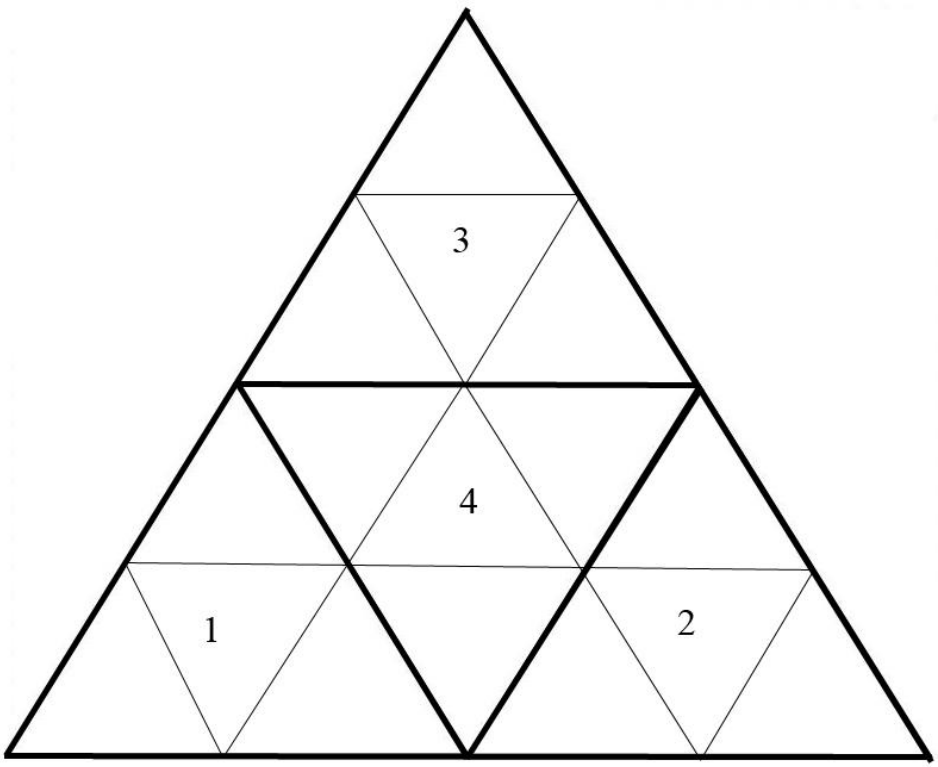
Why the Chaos Game Works

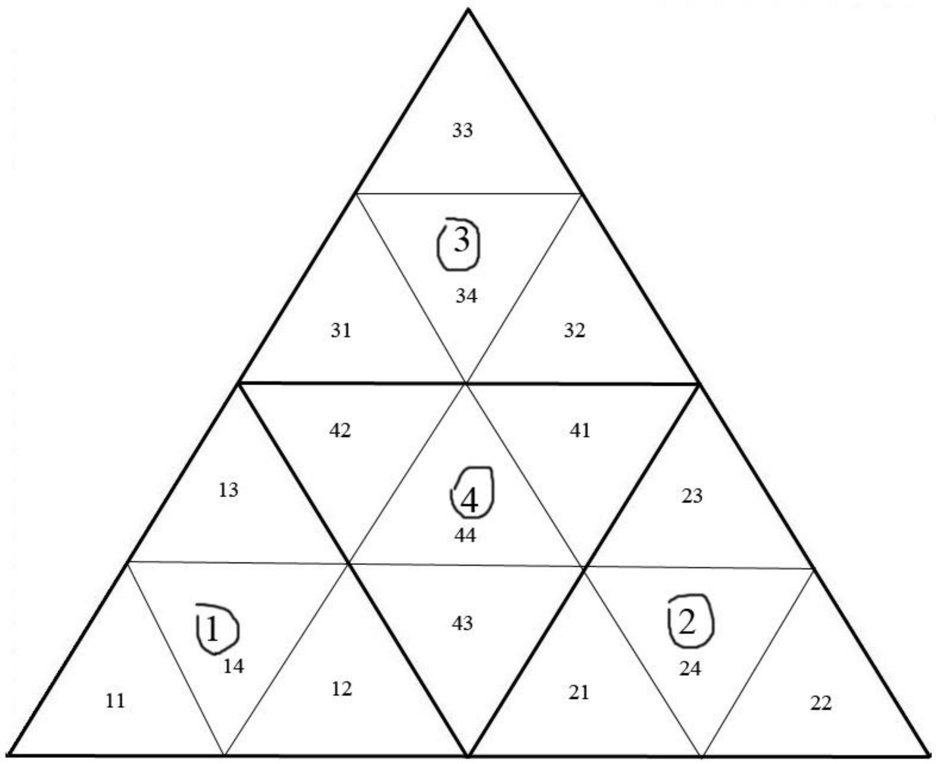
- Addresses:

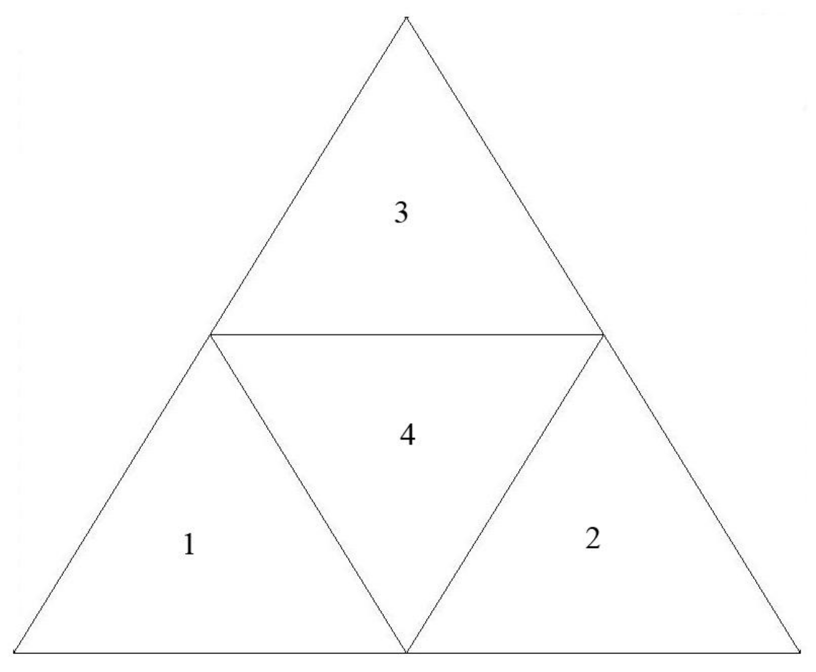


Why the Chaos Game Works

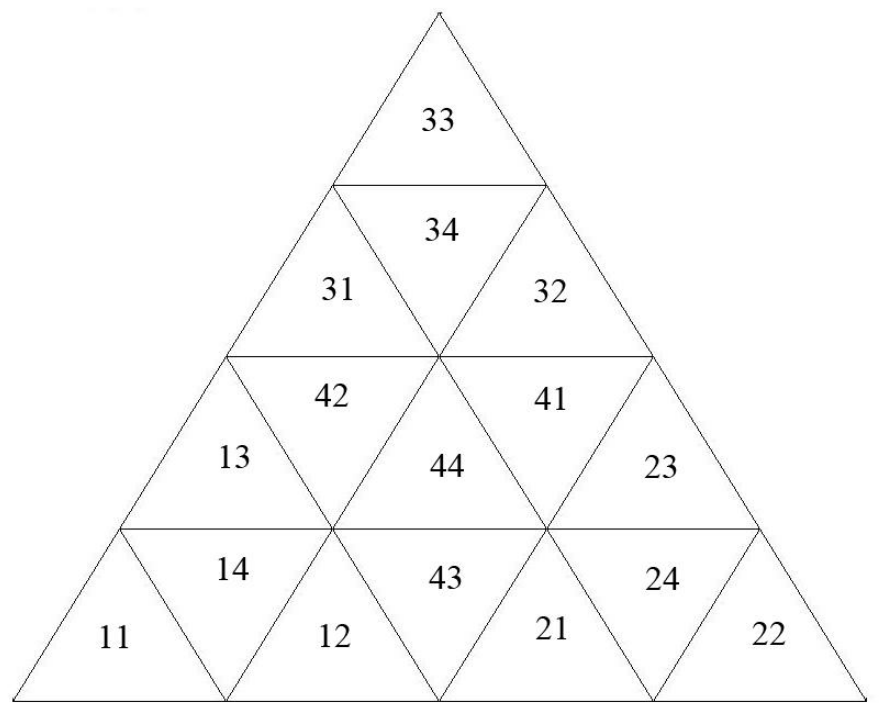
- Addresses:







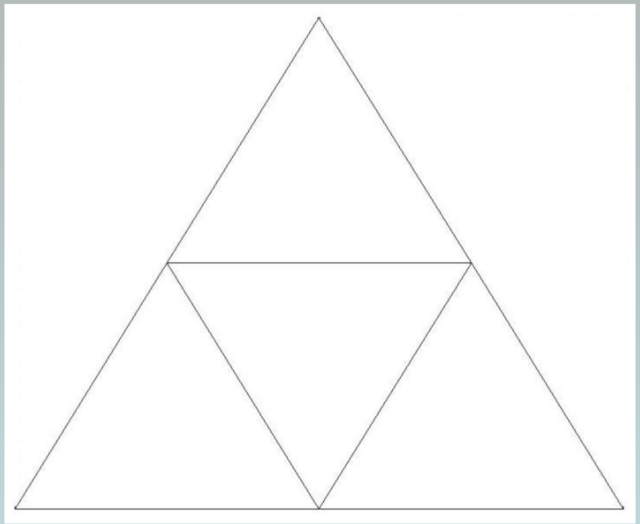
Address length 1



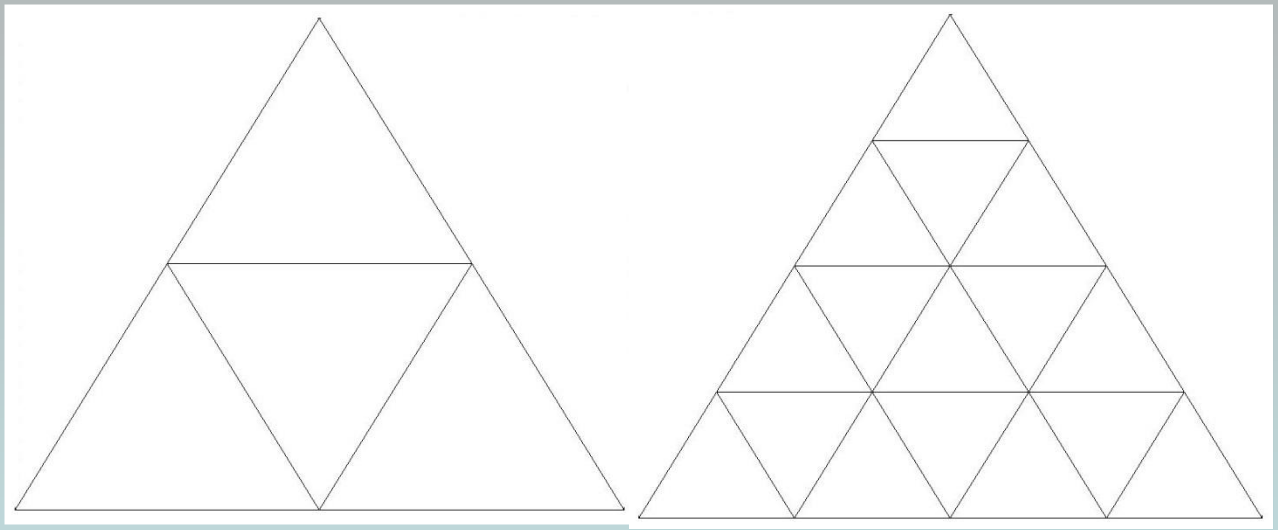
Address length 2

Address length 3

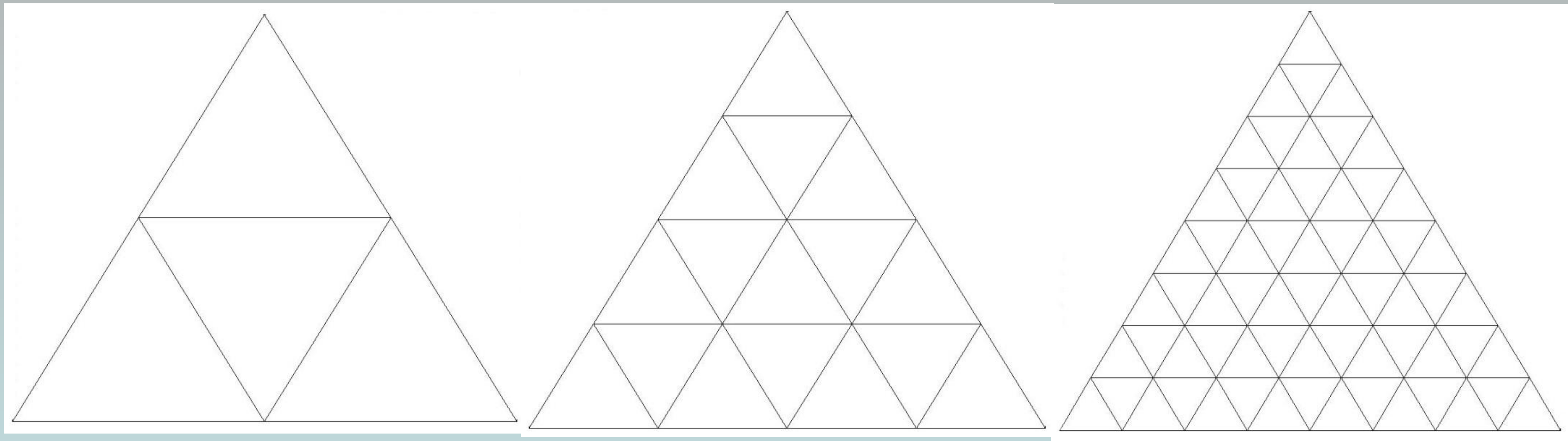
Address length 3



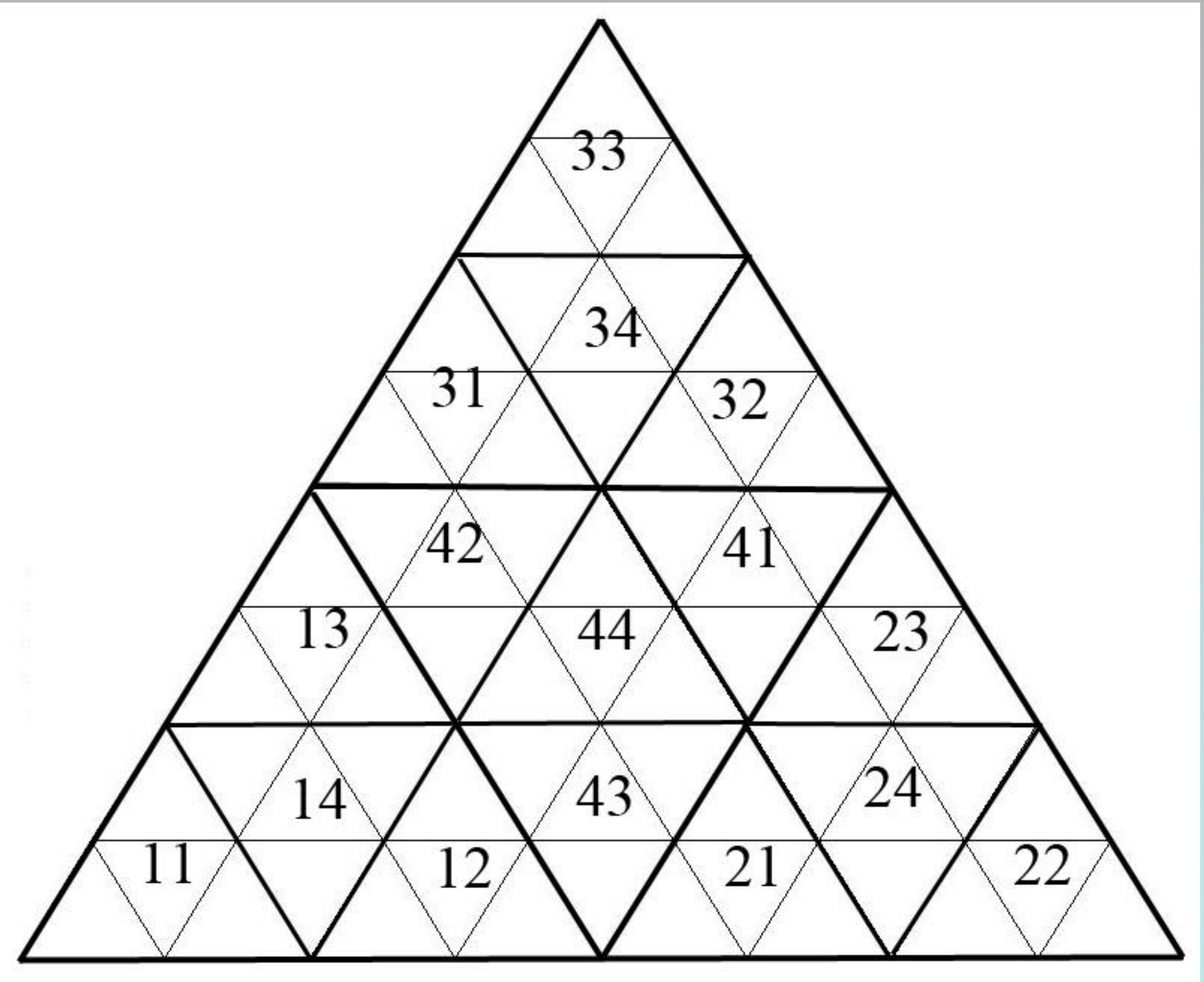
Address length 3

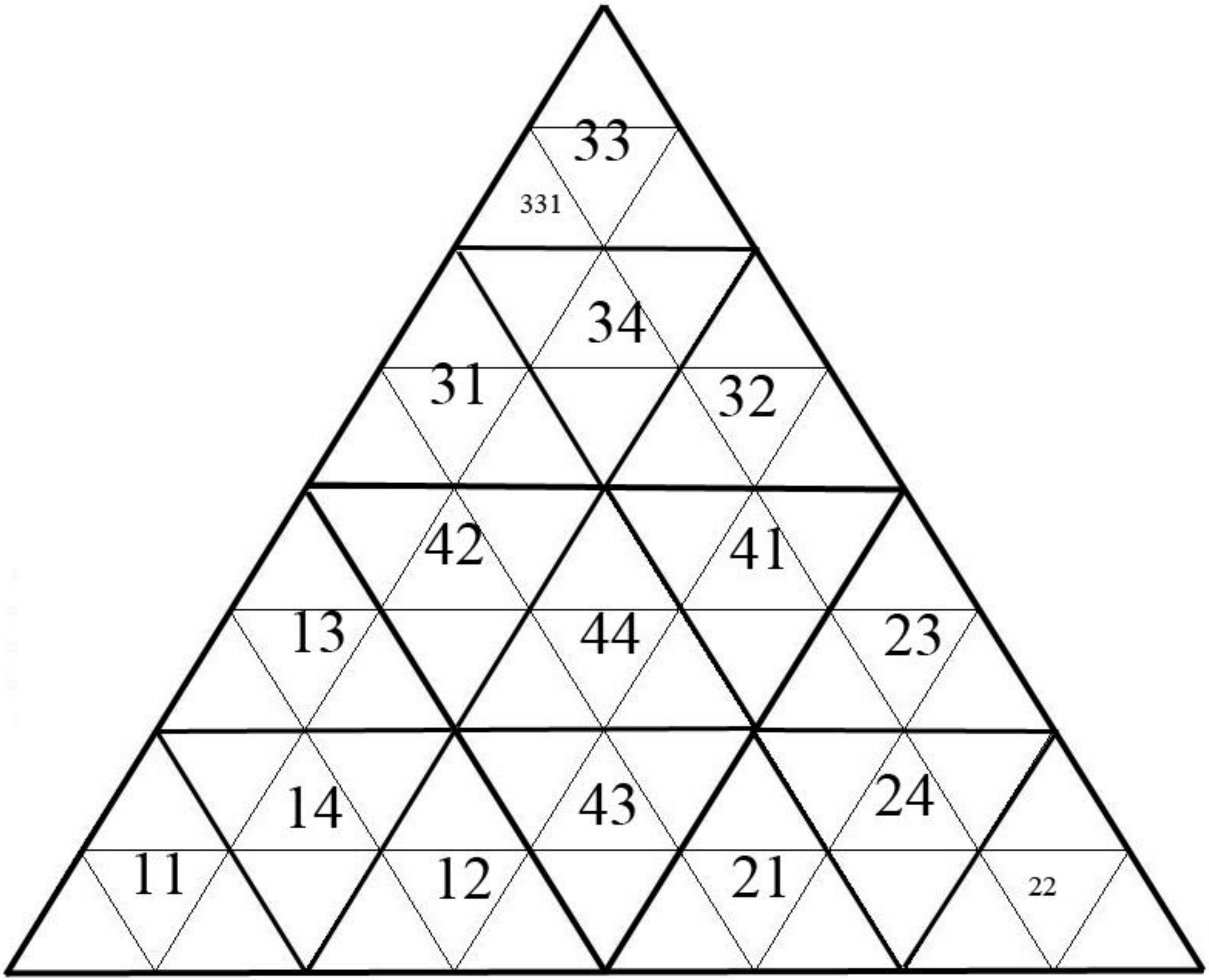


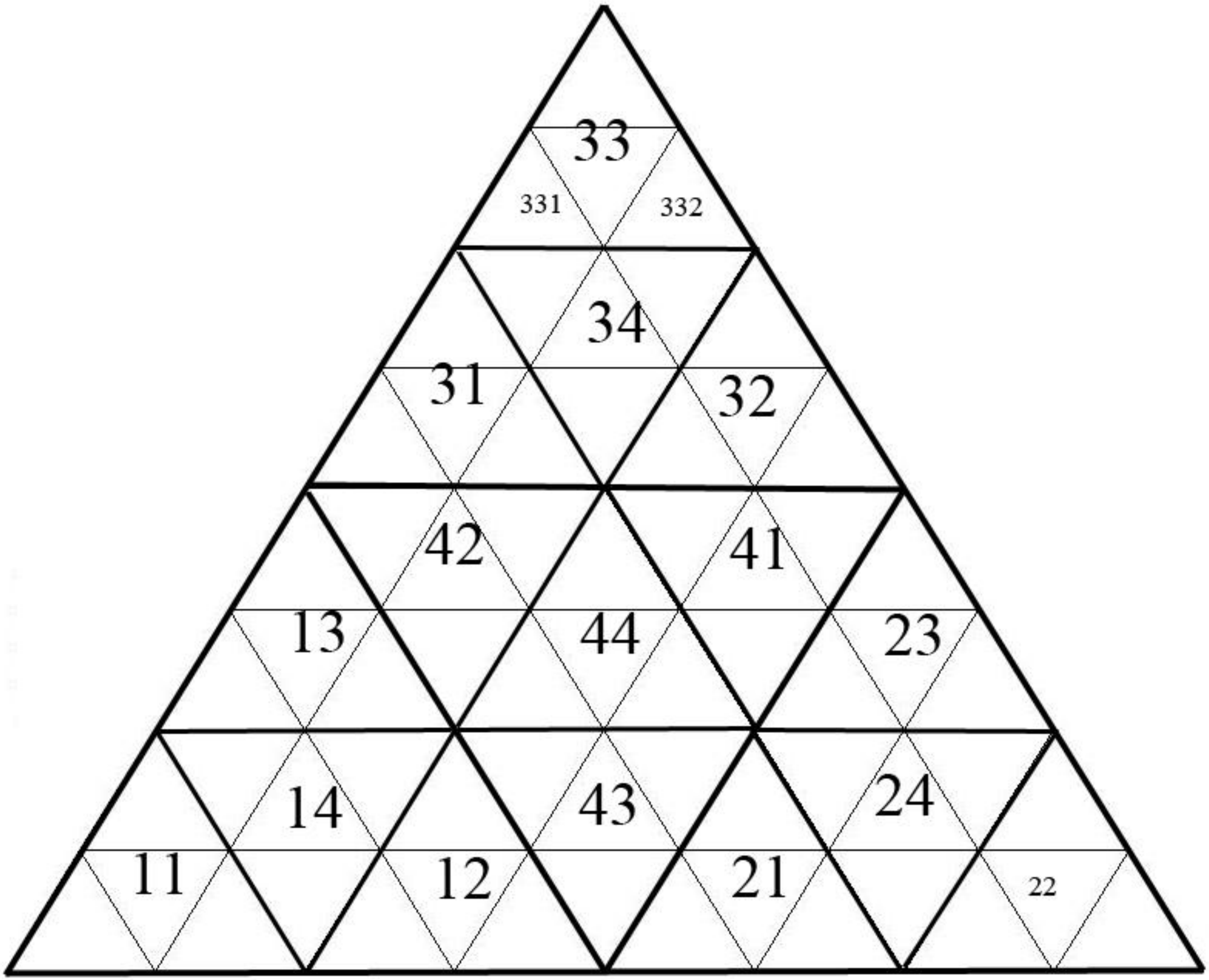
Address length 3

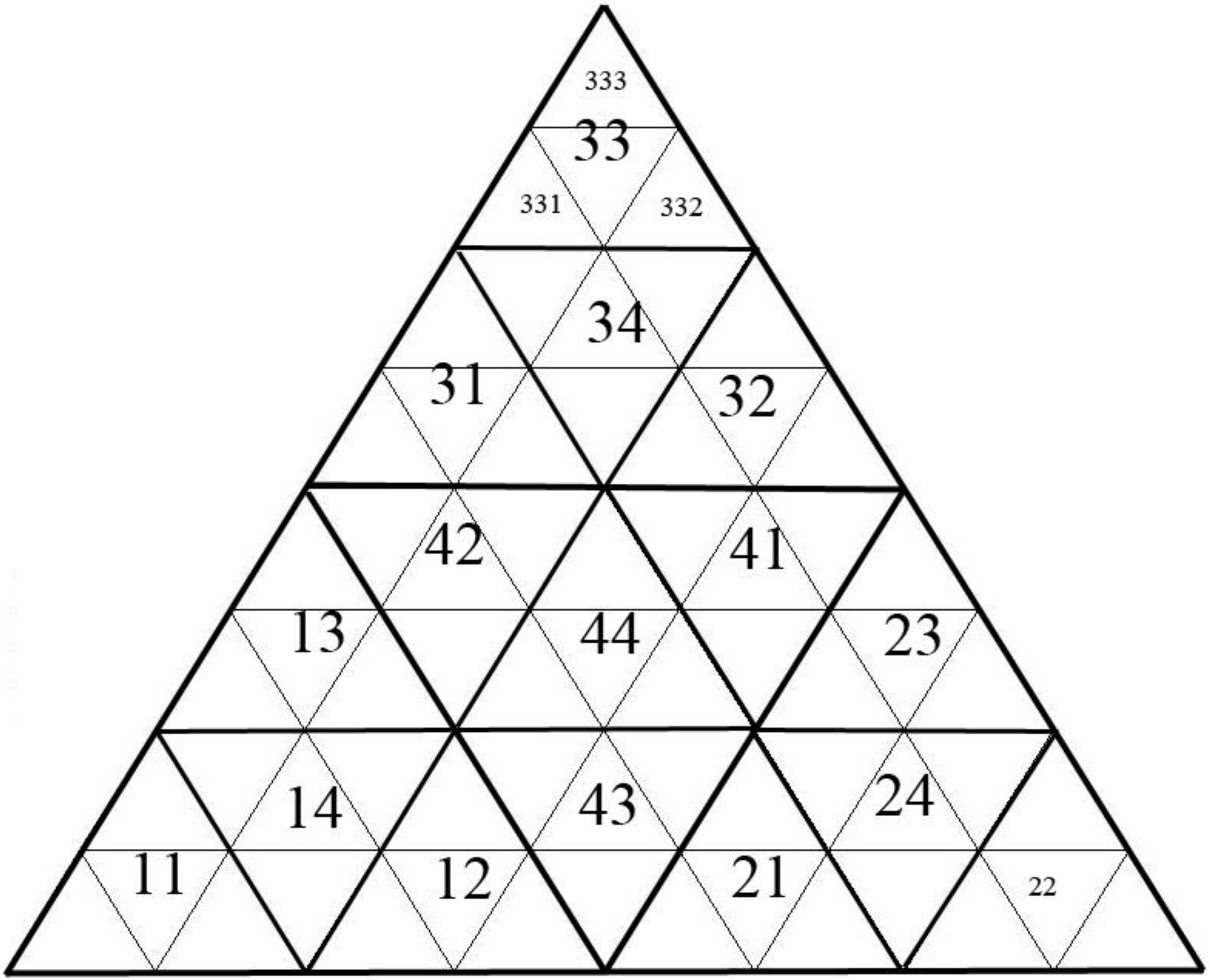


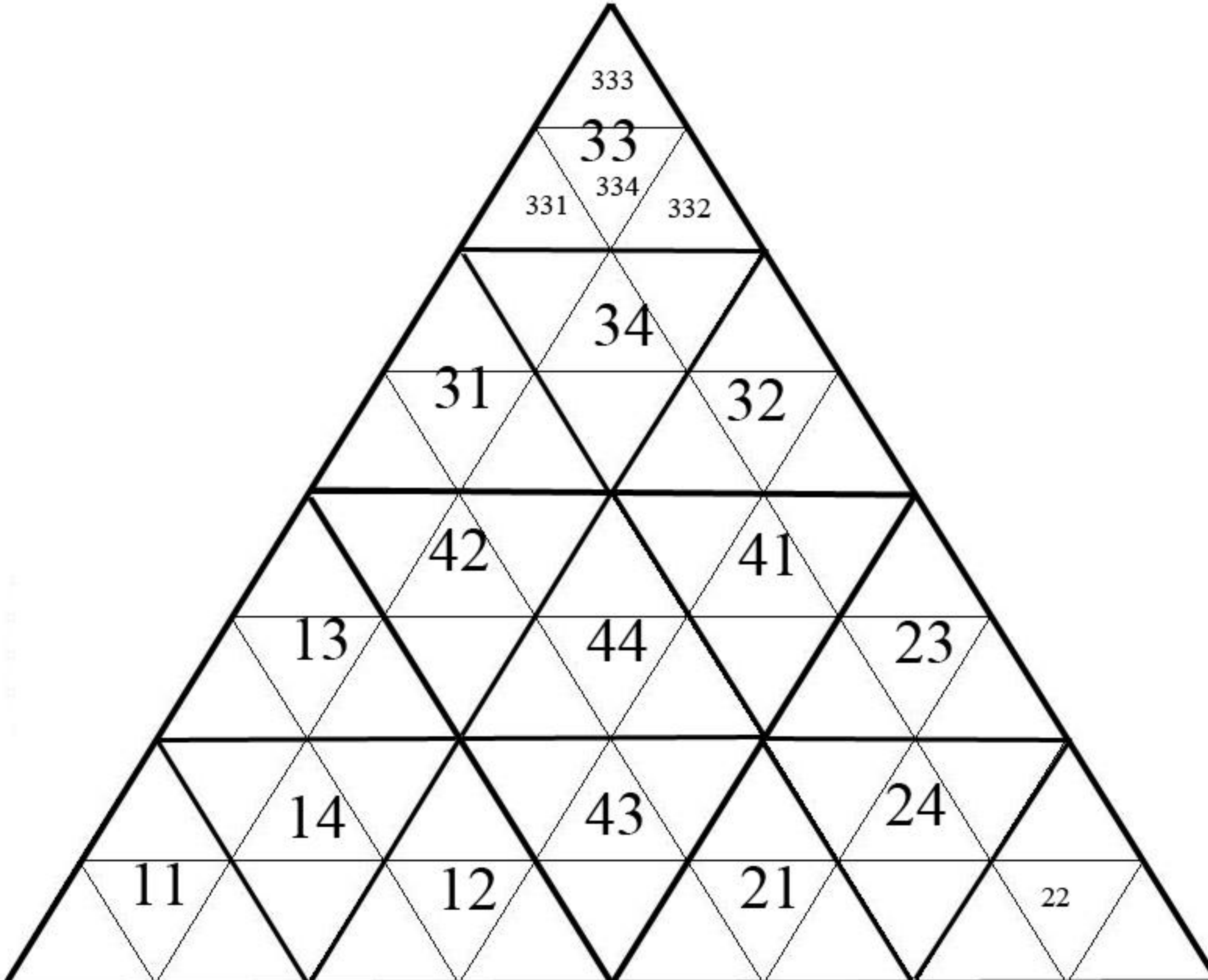
Address length 3



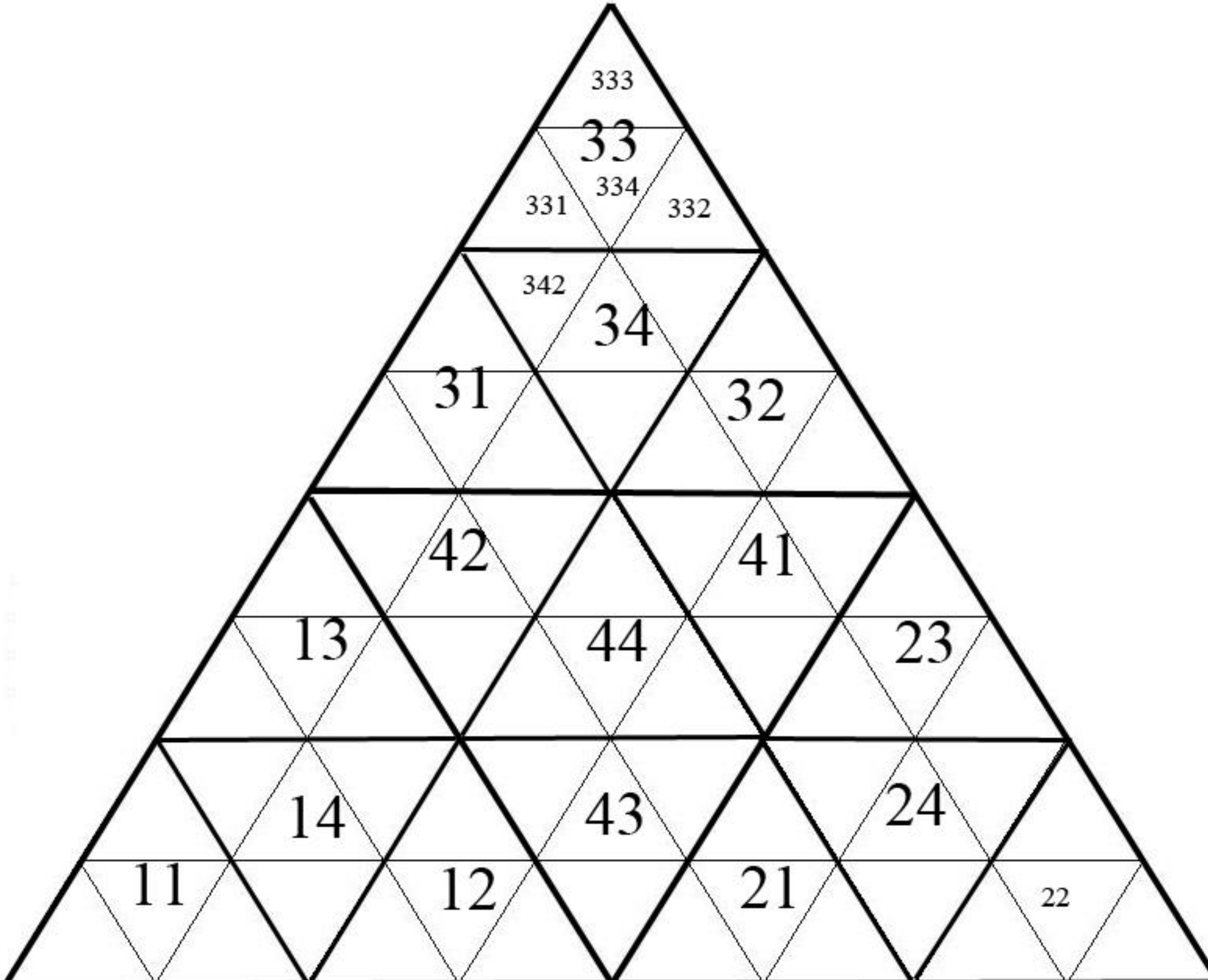

33
331
34
31
32
42
41
13
44
23
14
43
24
11
12
21
22

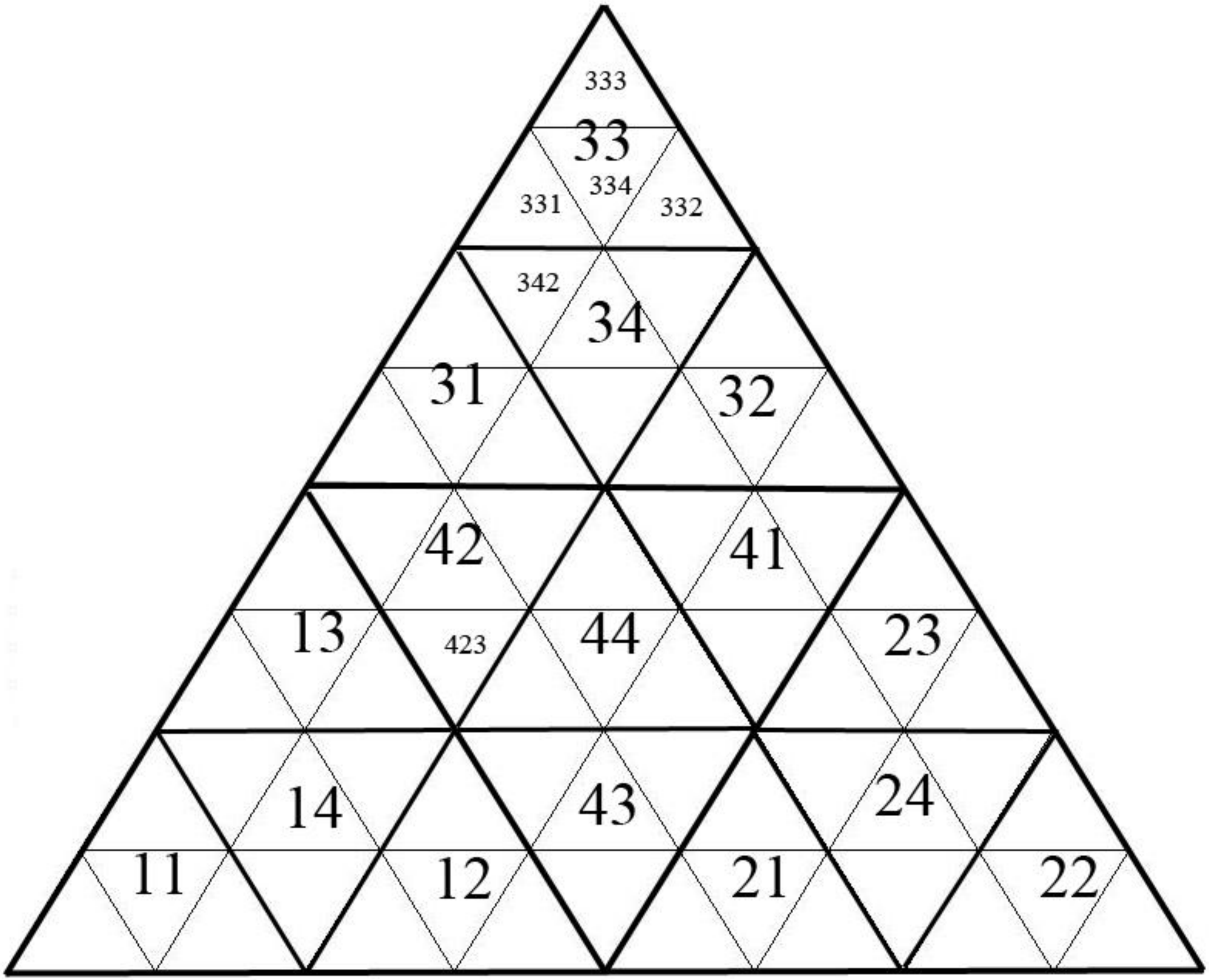

33
331
332
34
31
32
42
41
13
44
23
14
43
24
11
12
21
22

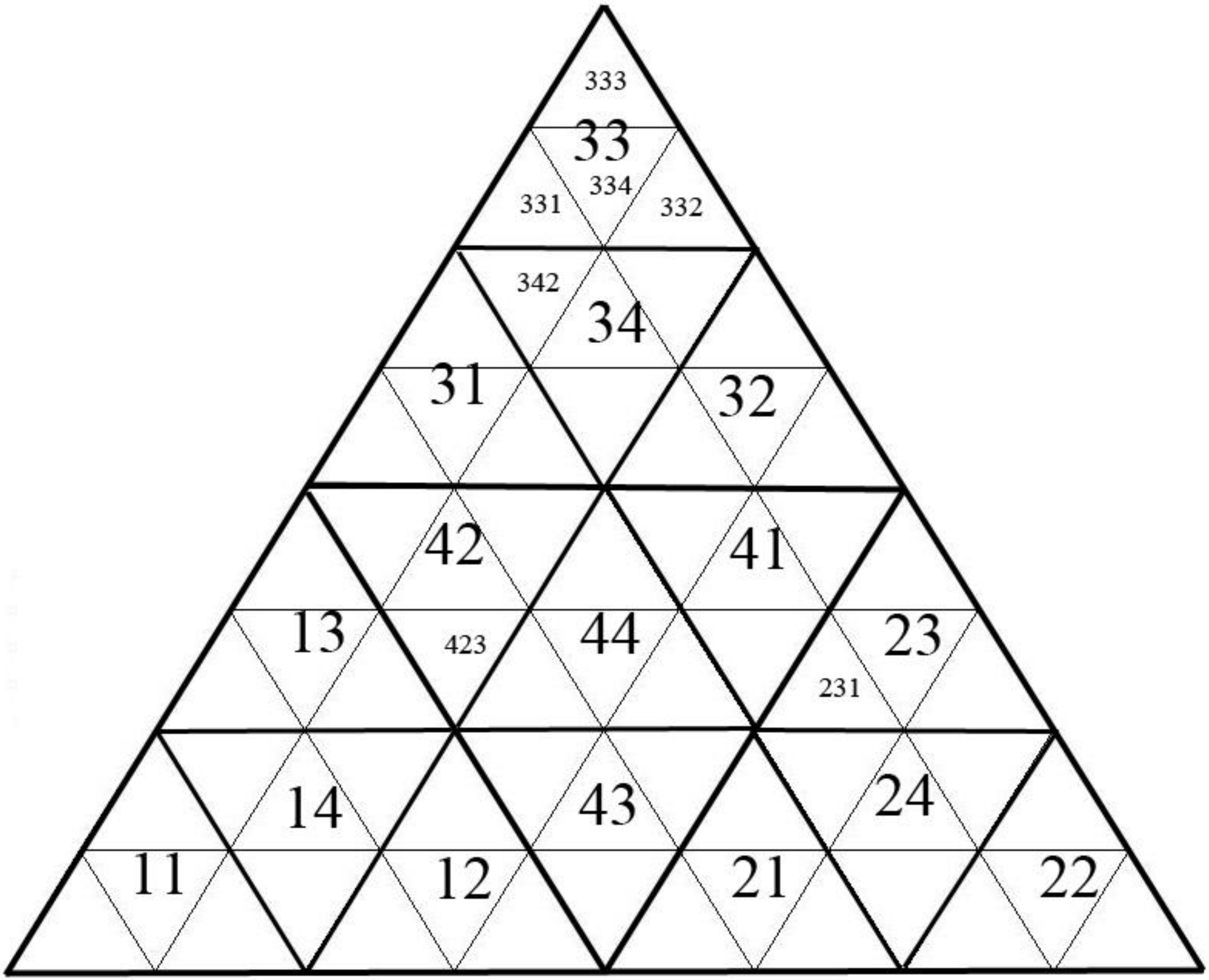

333
33
331
332
34
31
32
42
41
13
44
23
14
43
24
11
12
21
22



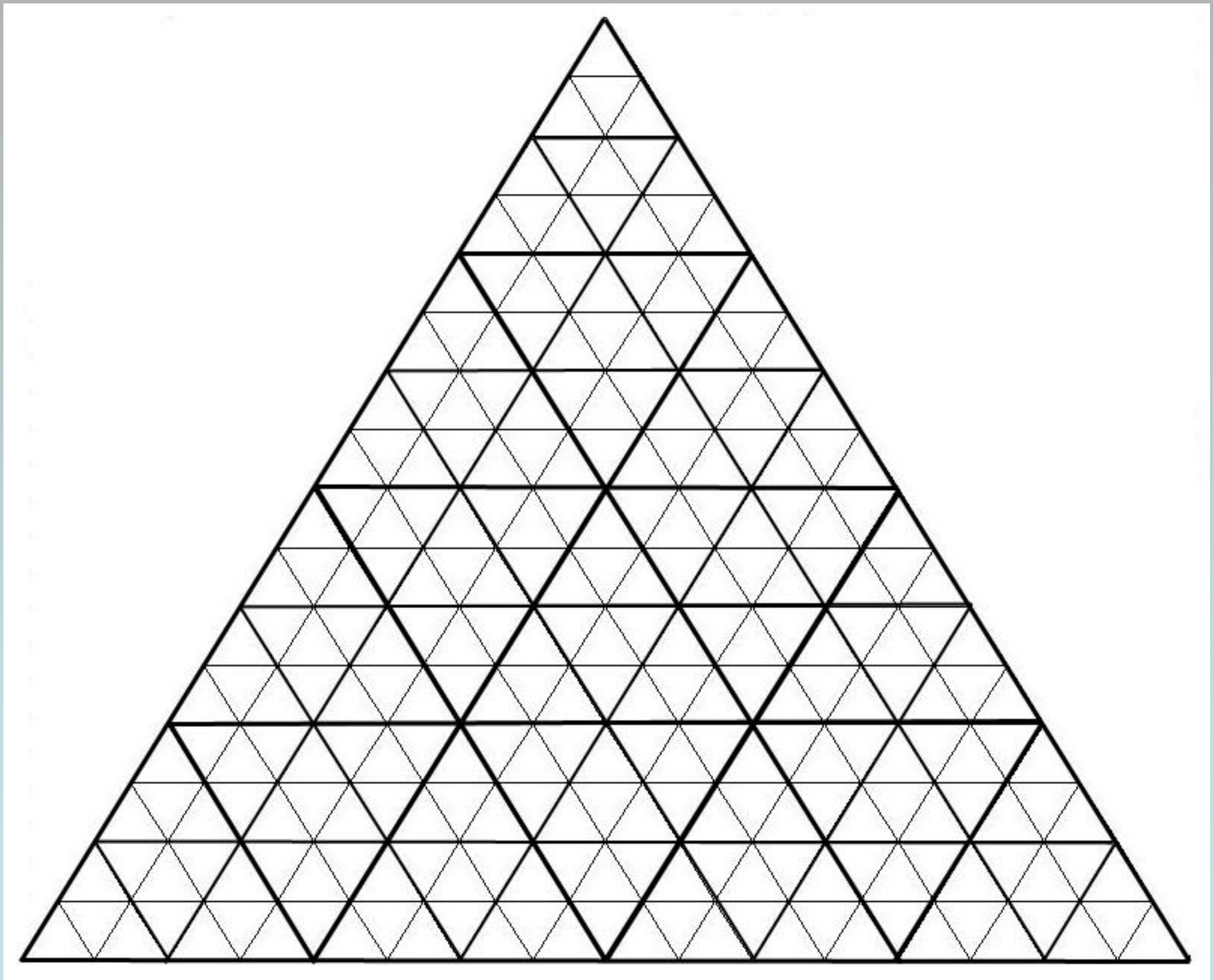
333
33
334
331
332
34
31
32
42
41
13
44
23
14
43
24
11
12
21
22


333
33
331
334
332
342
34
31
32
42
41
13
44
23
14
43
24
11
12
21
22

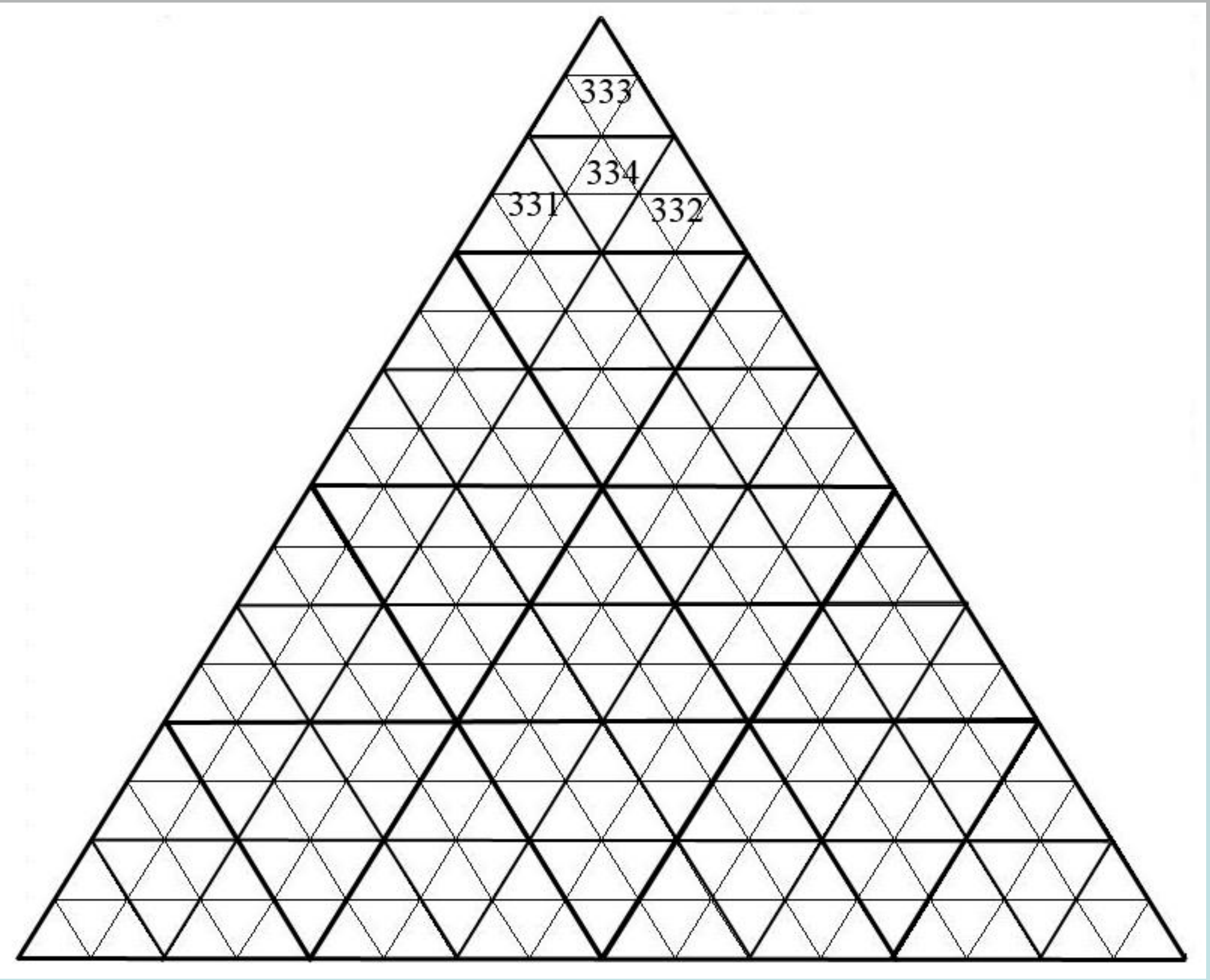

333
33
331
334
332
342
34
31
32
42
41
13
423
44
23
14
43
24
11
12
21
22


333
33
331
334
332
342
34
31
32
42
41
13
423
44
23
231
14
43
24
11
12
21
22

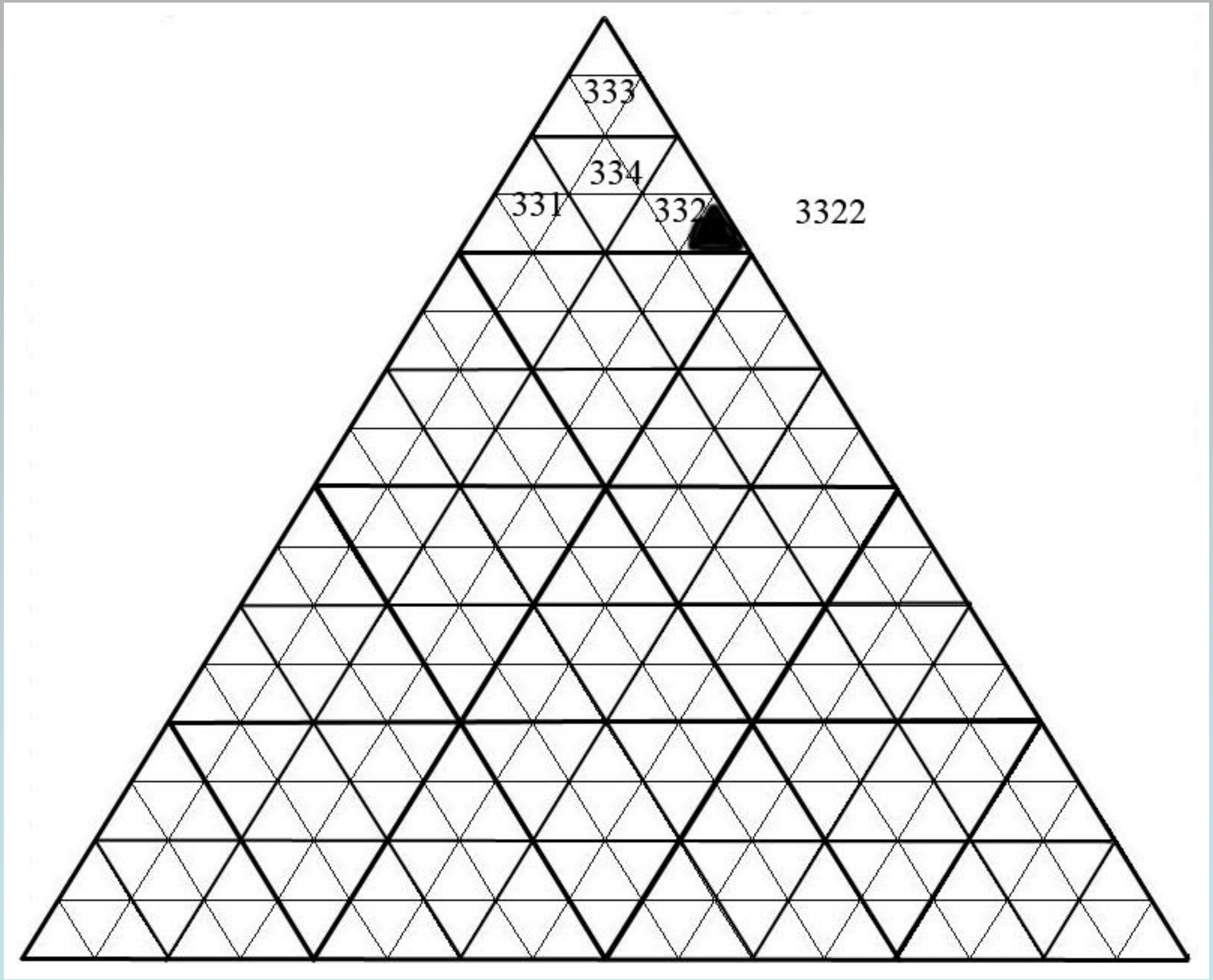
Address length 4



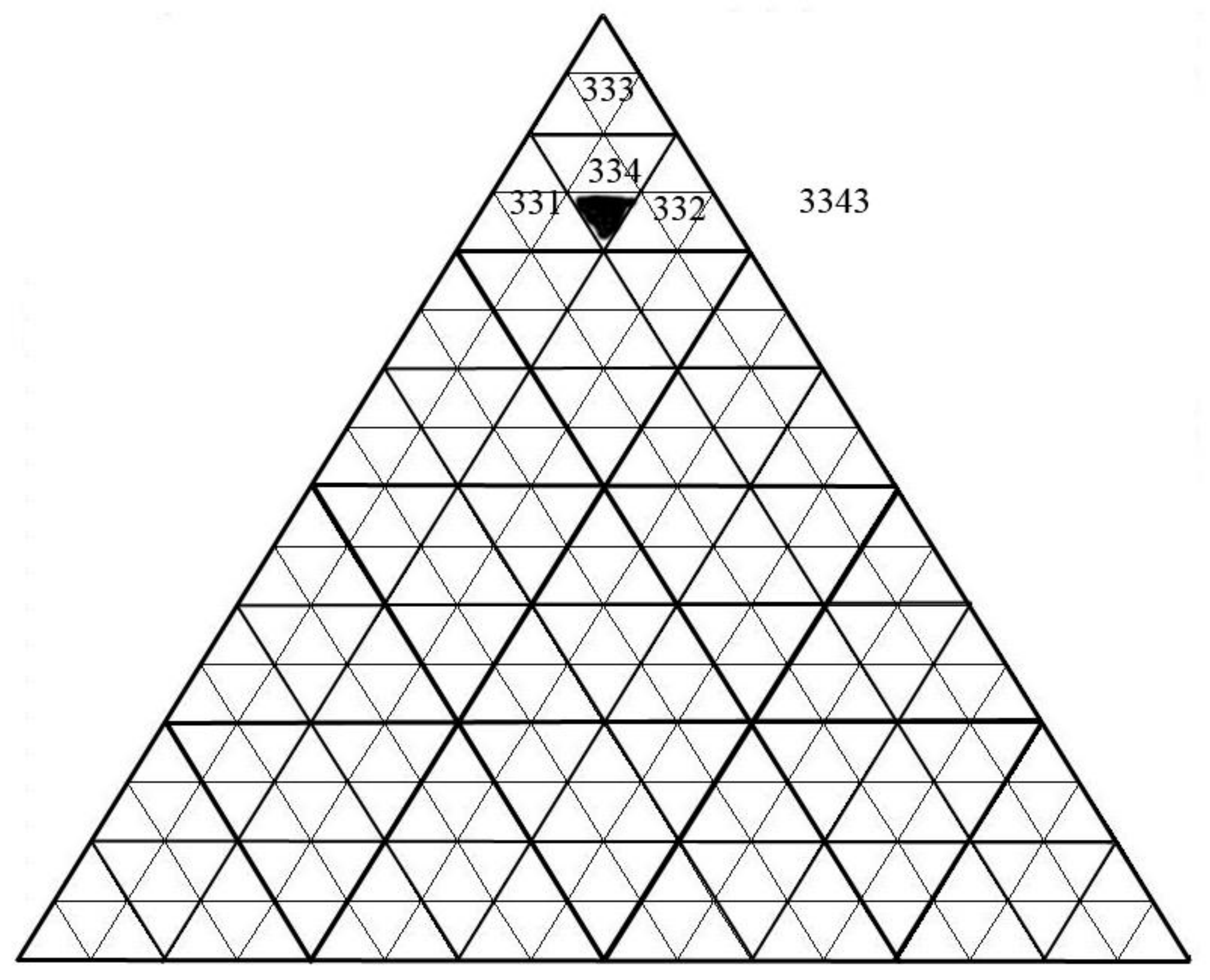
Address length 4

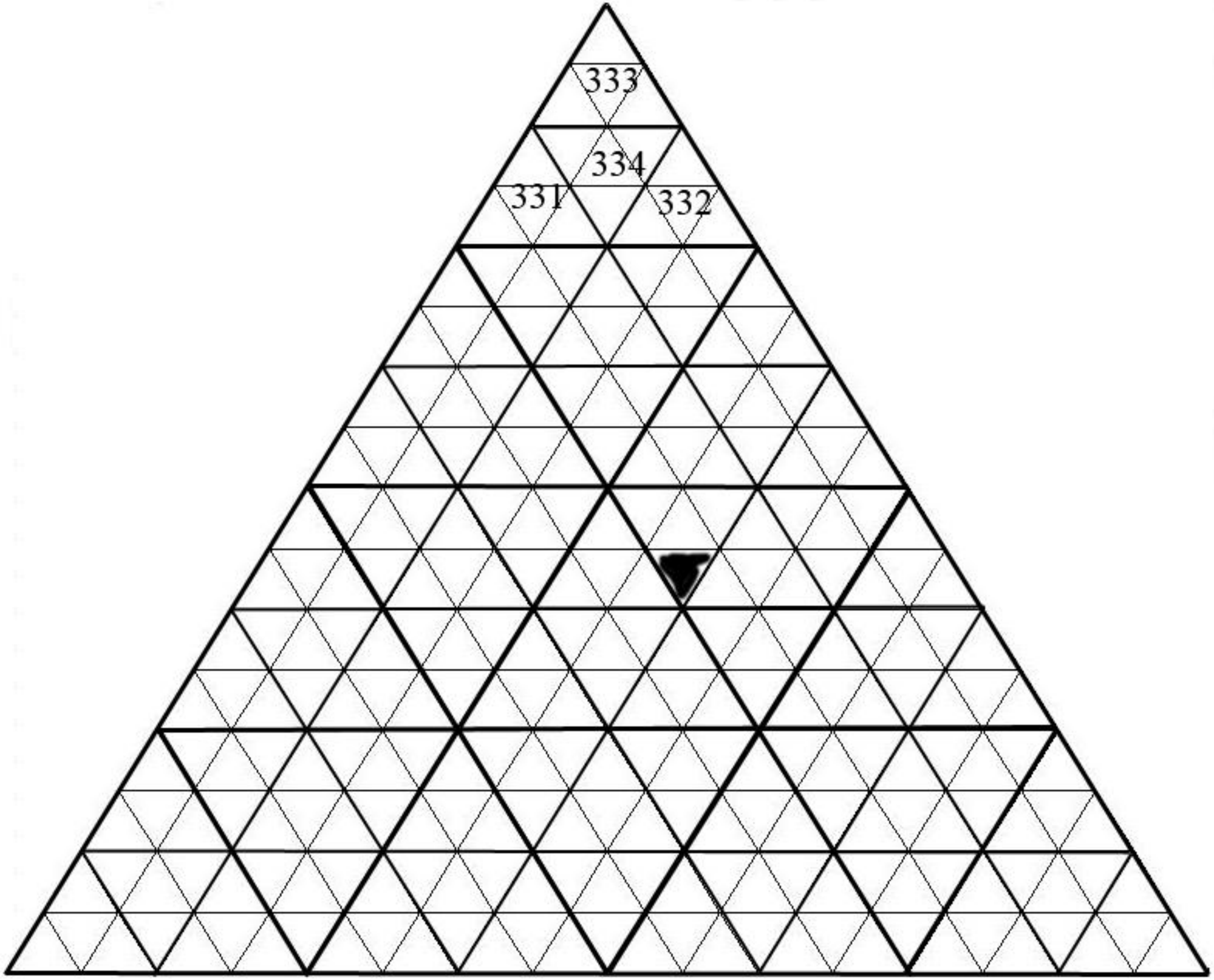


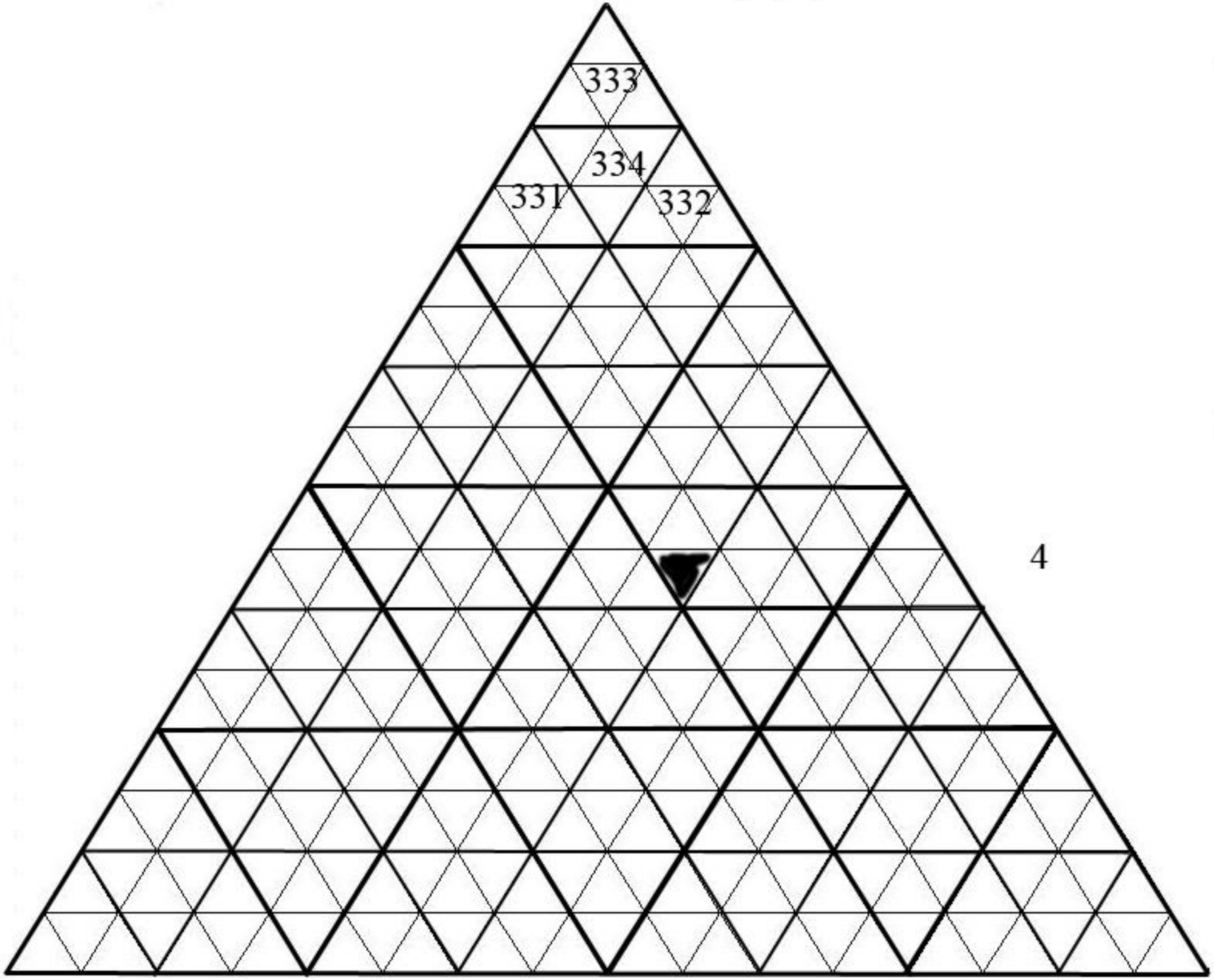
Address length 4

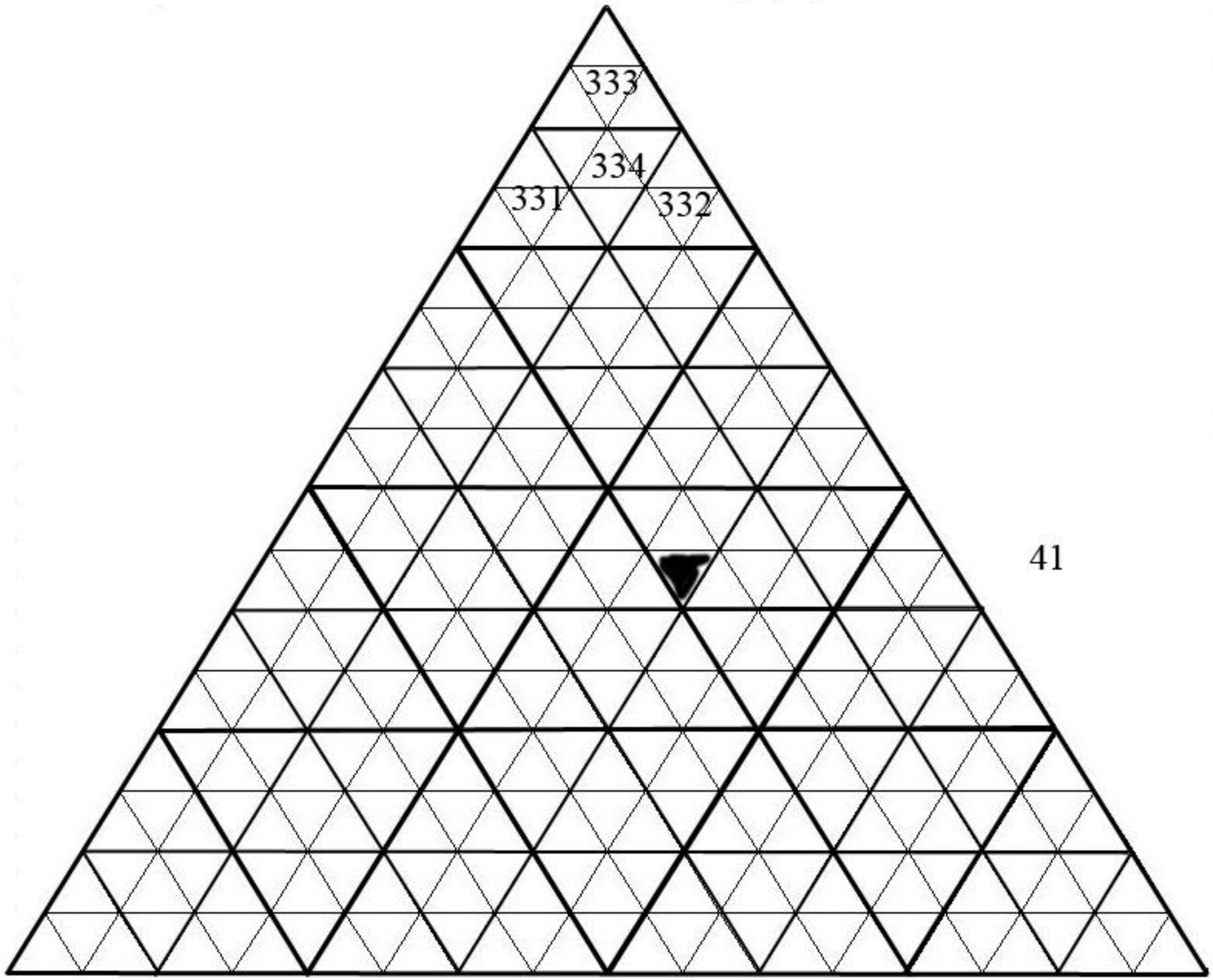


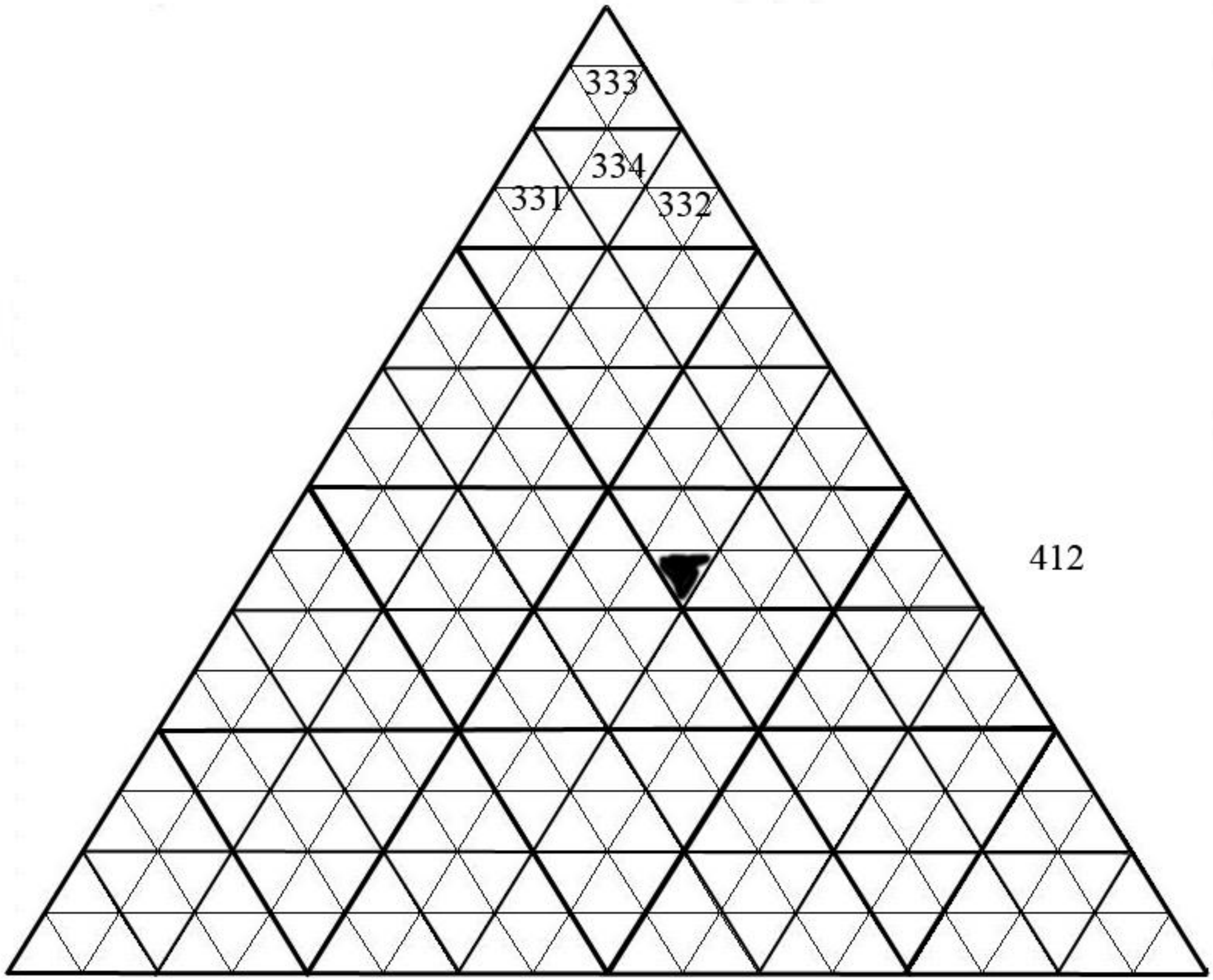
Address length 4

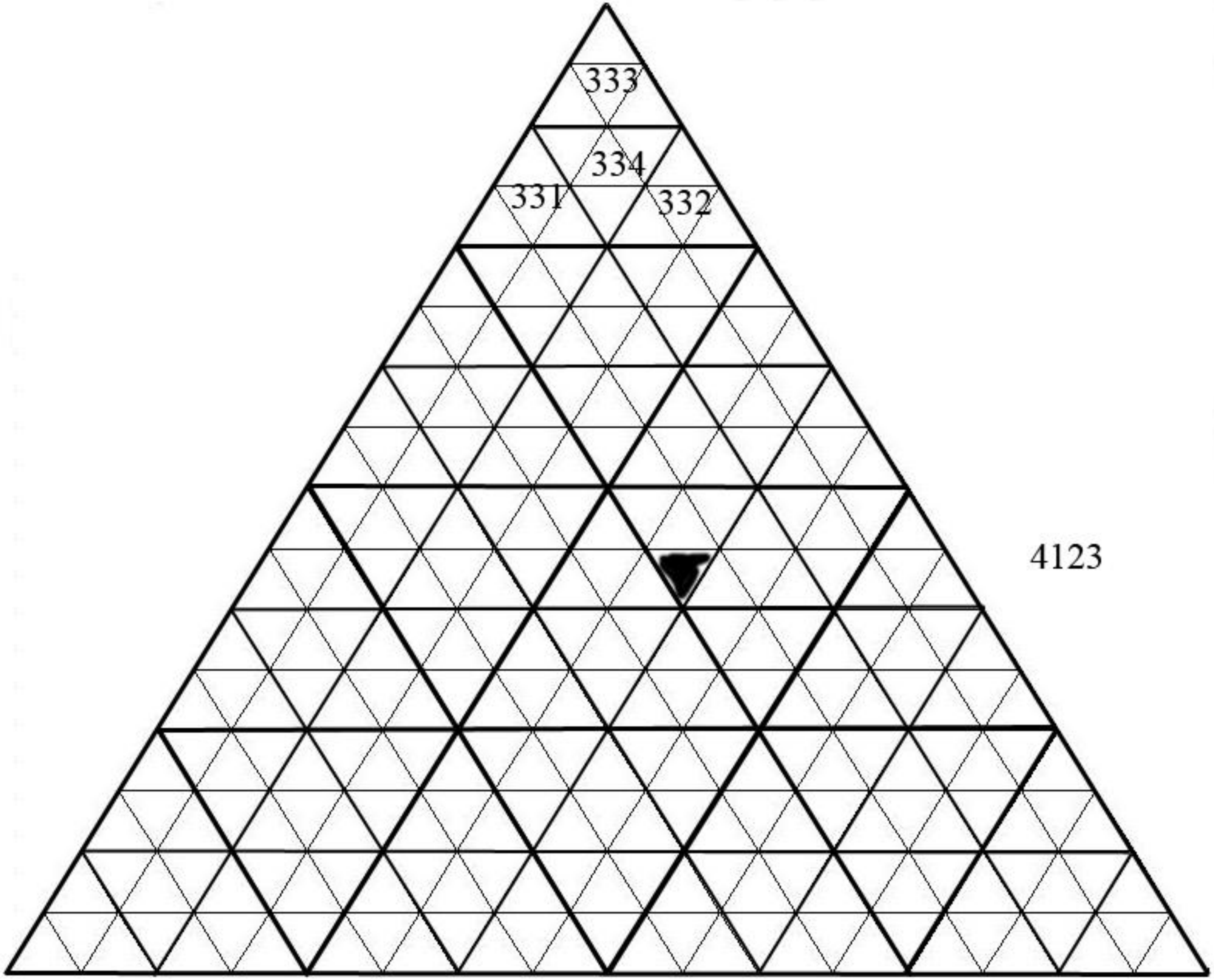


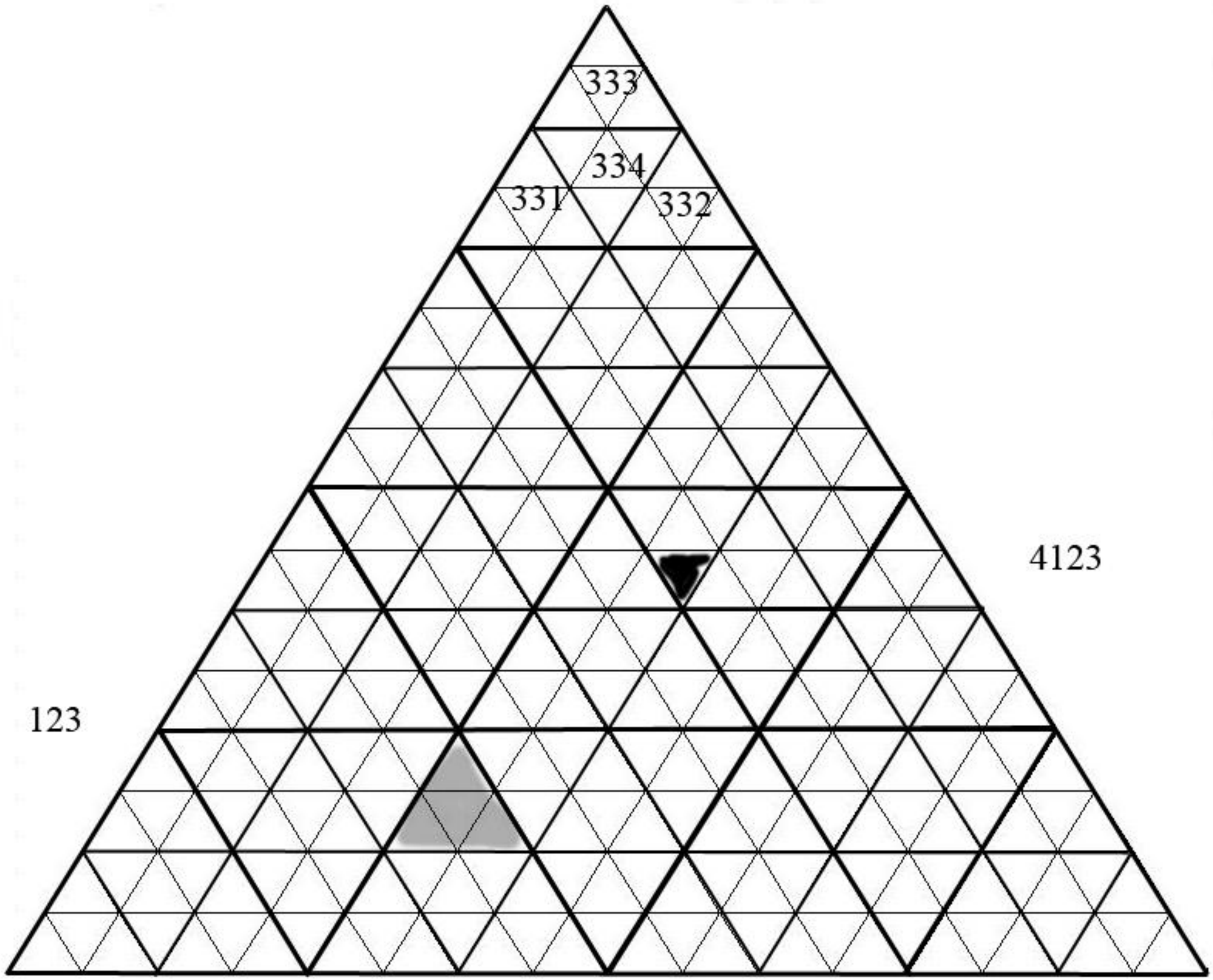

333
334
331
332


333
334
331
332
4


333
334
331
332
41


333
334
331
332
412

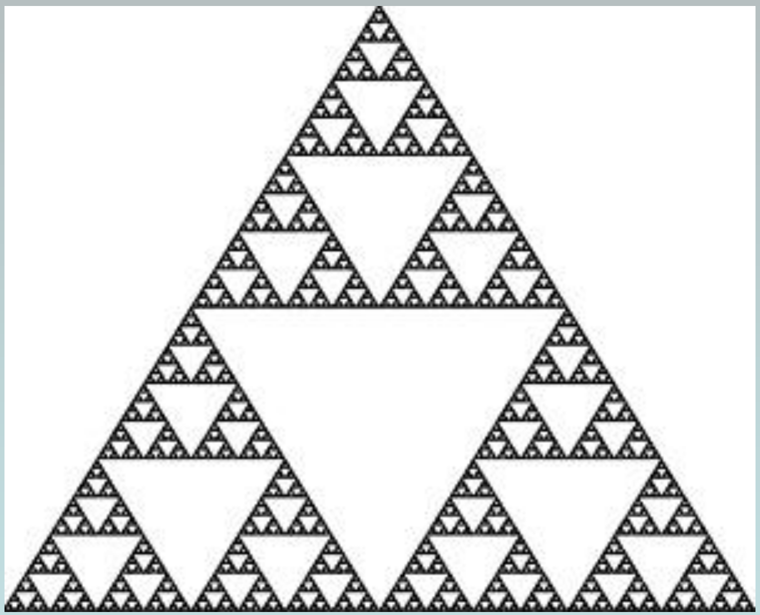

333
334
331
332
4123

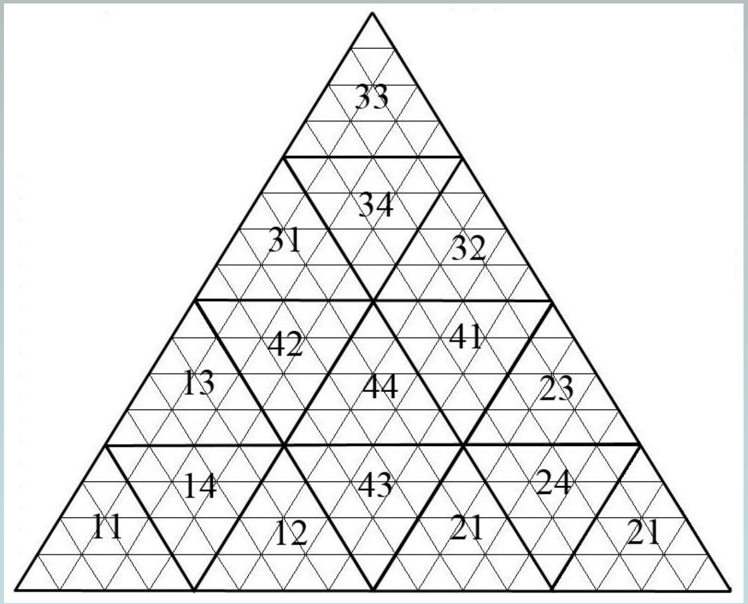

333
334
331
332
4123
123

What is Sierpinski’s Triangle?

What is Sierpinski's Triangle?

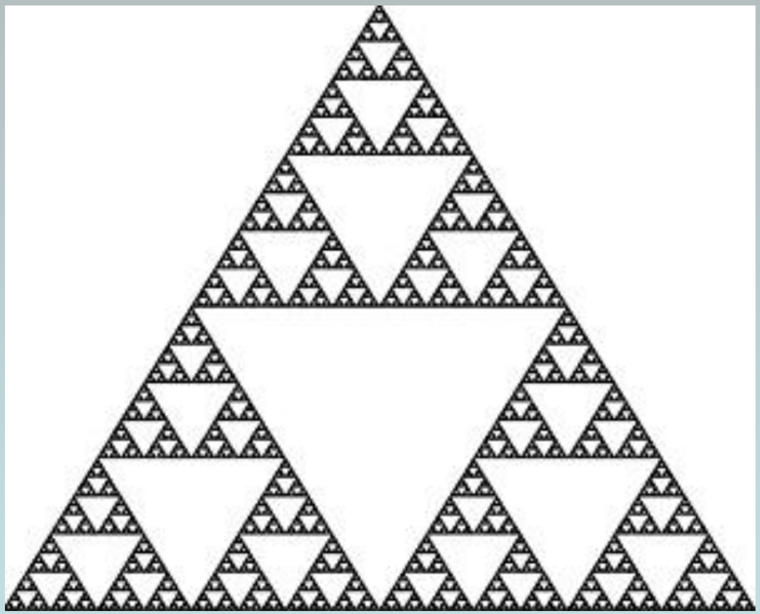
All regions without a '4' in their address:

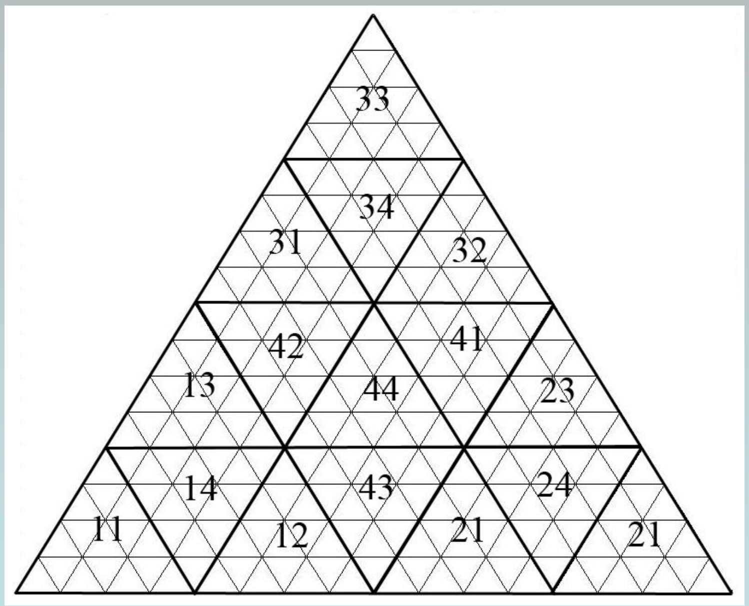




What is Sierpinski's Triangle?

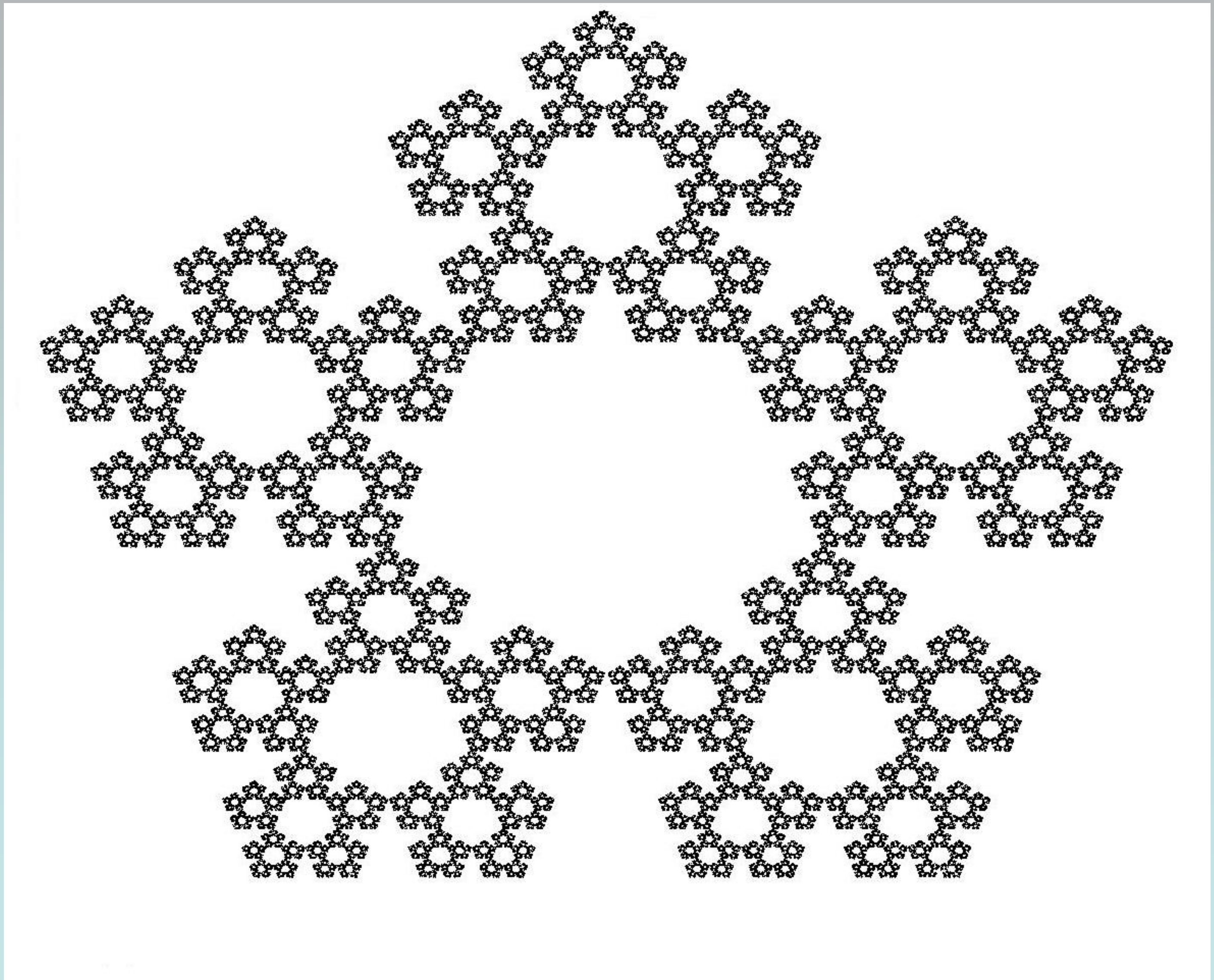
All regions without a '4' in their address:



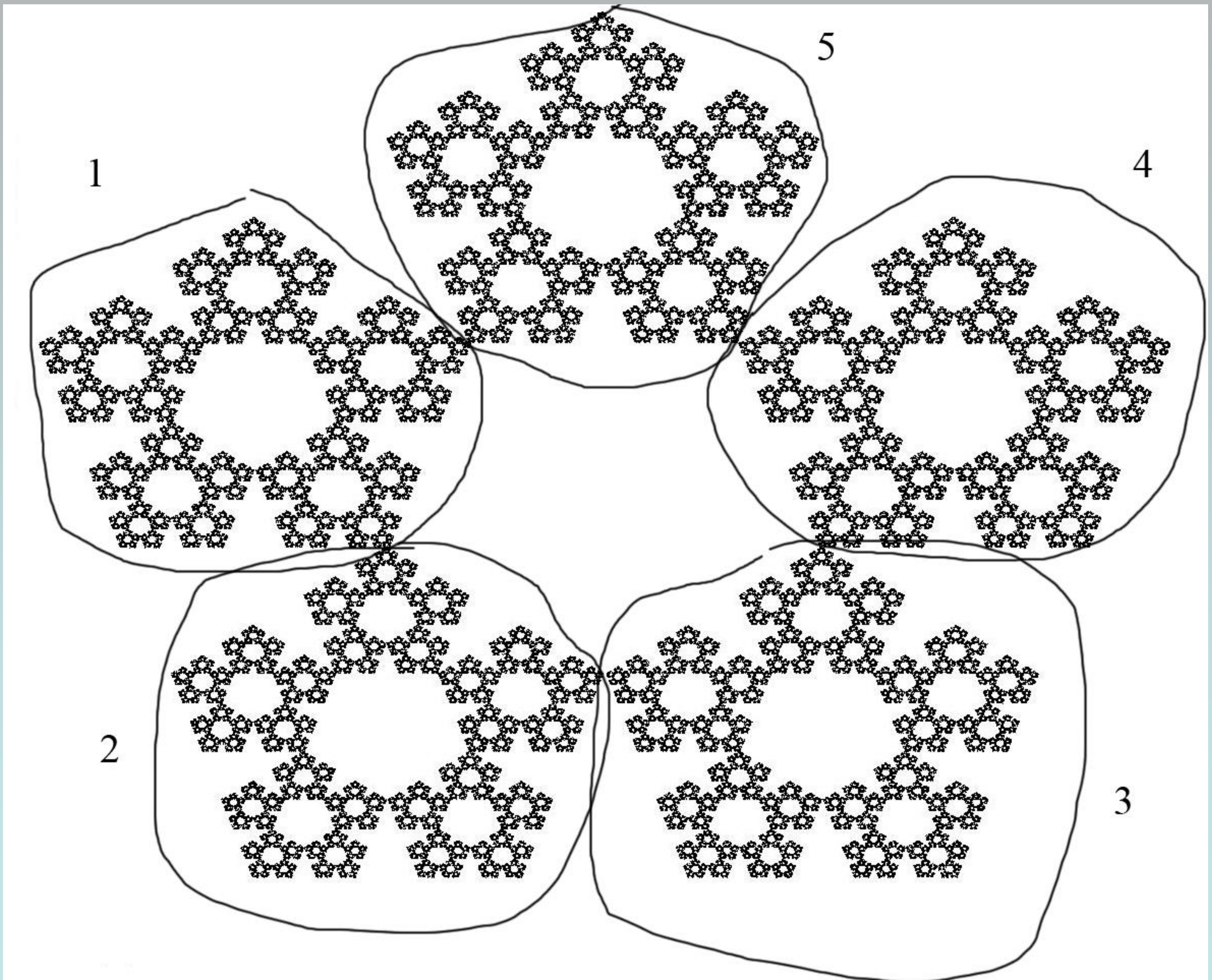


So we can draw the Sierpinski triangle if we put a dot in every address region that doesn't have a 4

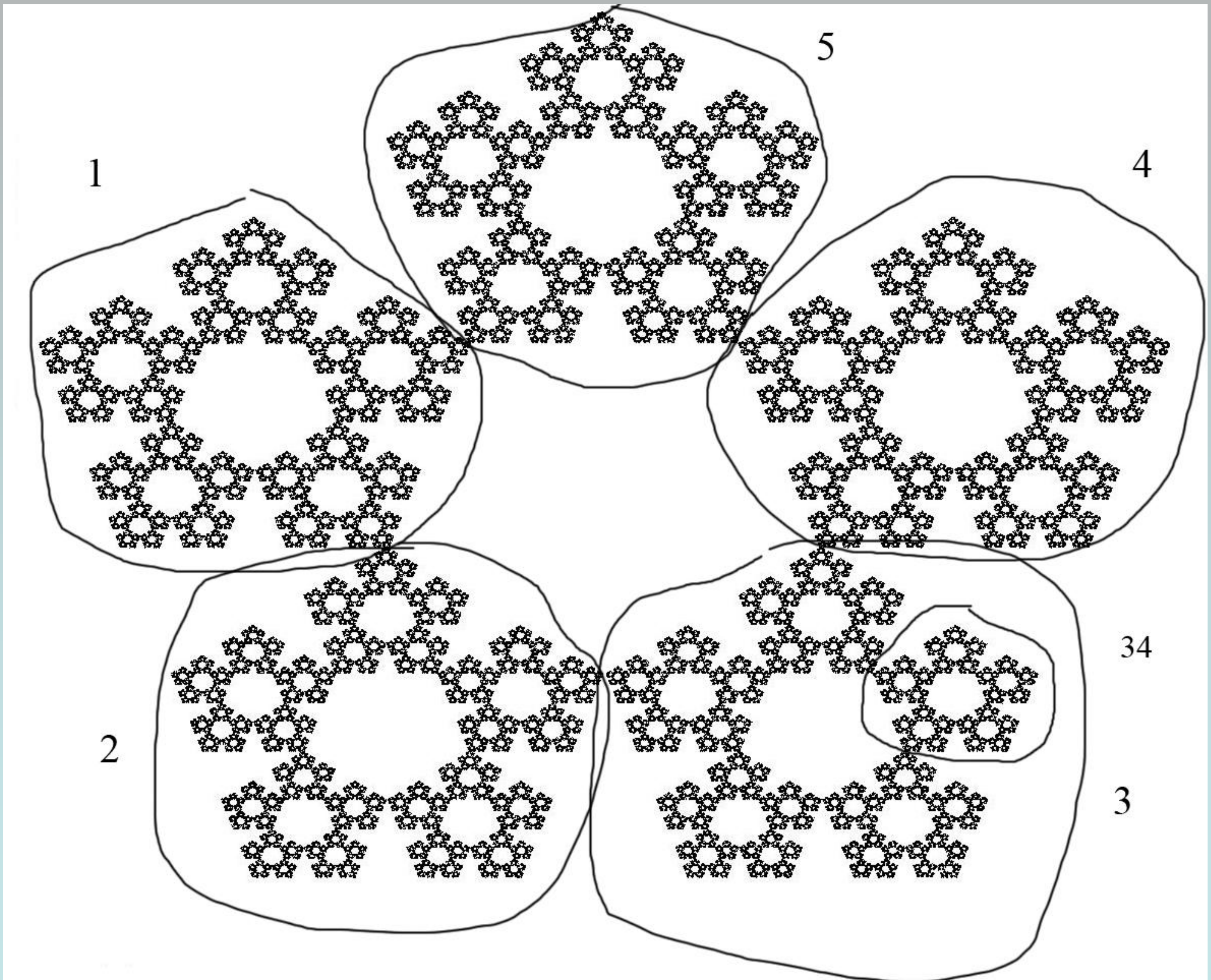
All fractals have an address system you can use to label parts of the fractal



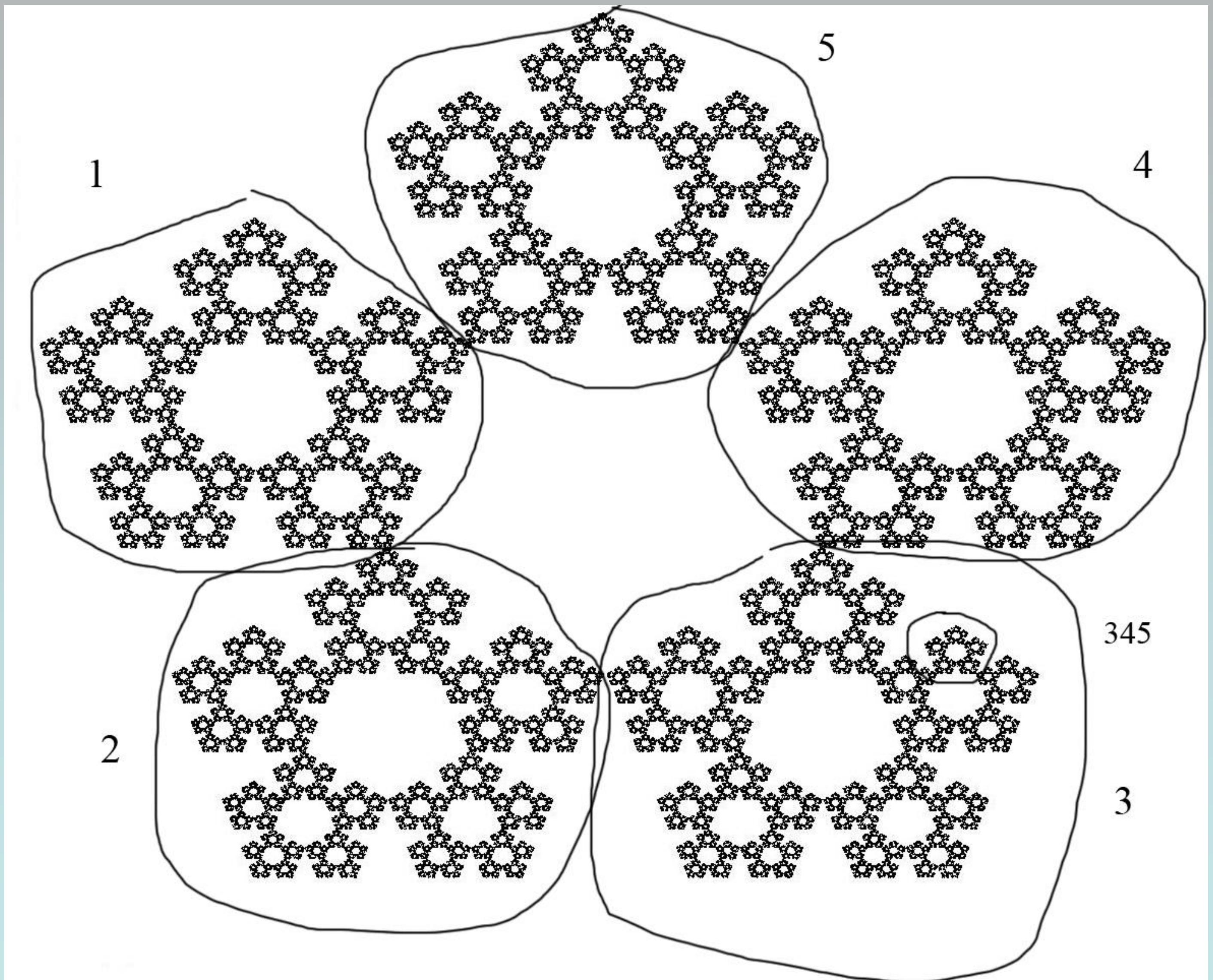
All fractals have an address system you can use to label parts of the fractal



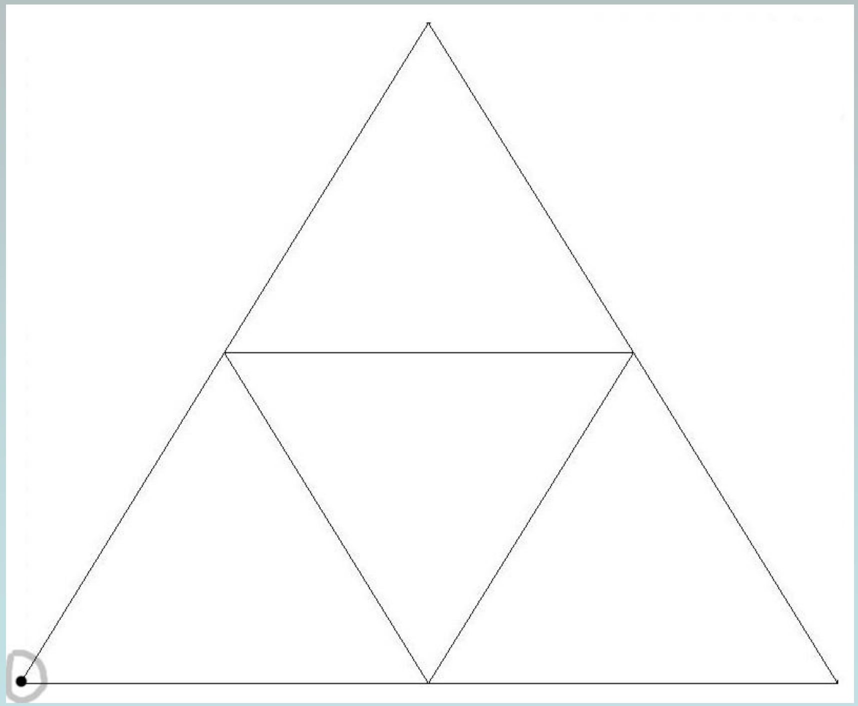
All fractals have an address system you can use to label parts of the fractal



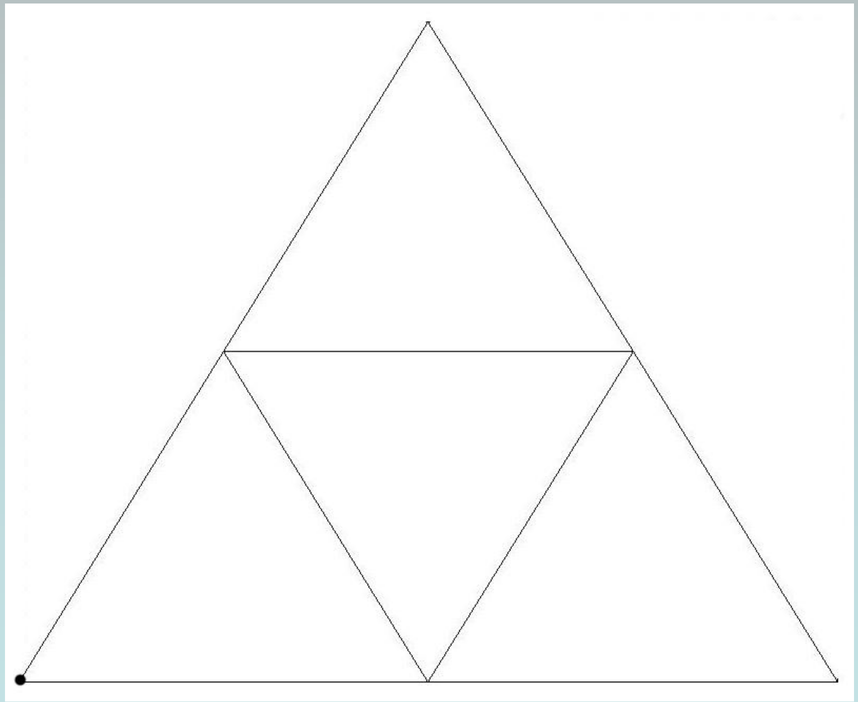
All fractals have an address system you can use to label parts of the fractal



Let's begin the Sierpinski Chaos game at the bottom left corner

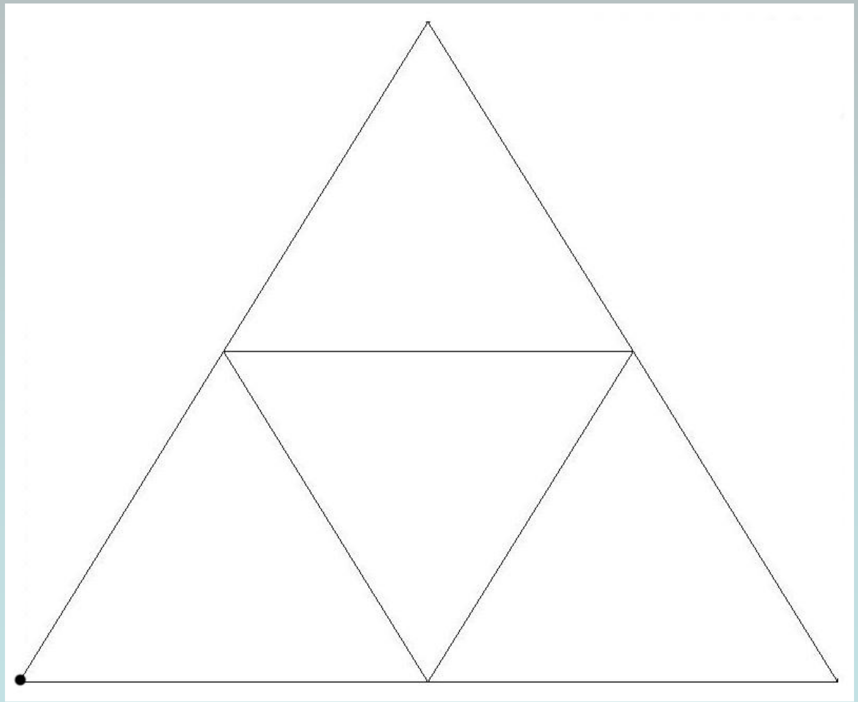


Let's begin the Sierpinski Chaos game at the bottom left corner

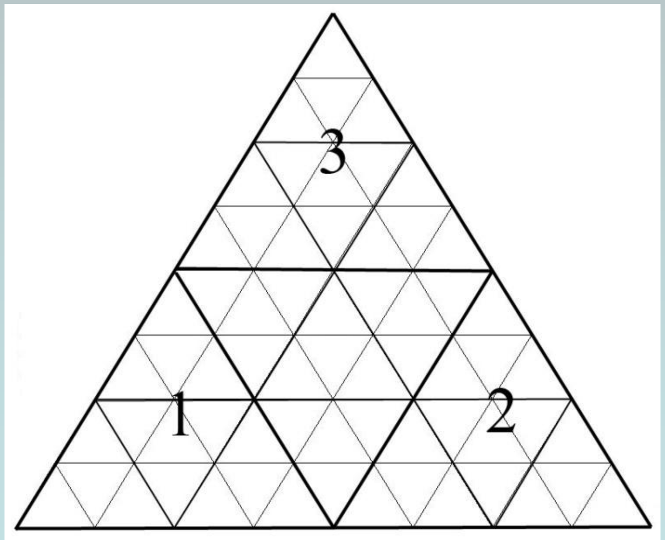


Address of this point is

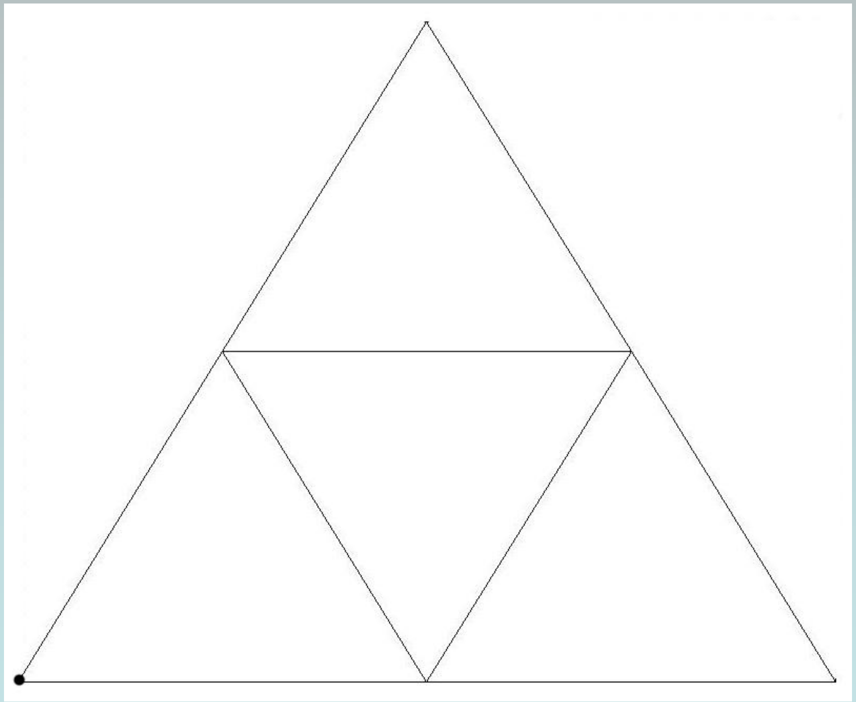
Let's begin the Sierpinski Chaos game at the bottom left corner



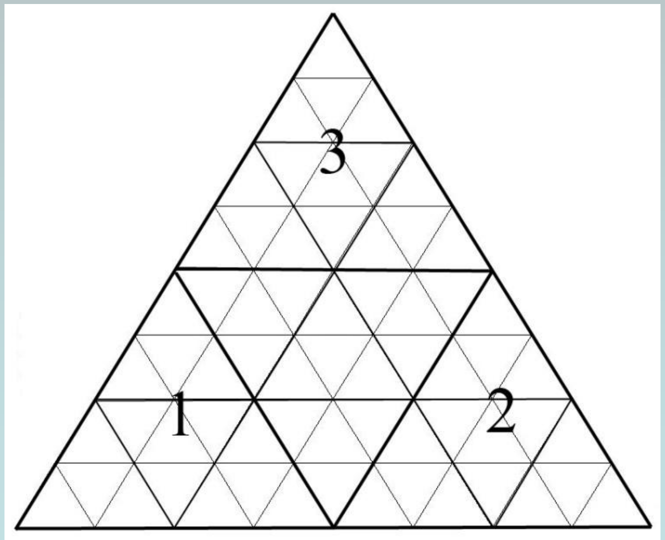
Address of this point is



Let’s begin the Sierpinski Chaos game at the bottom left corner

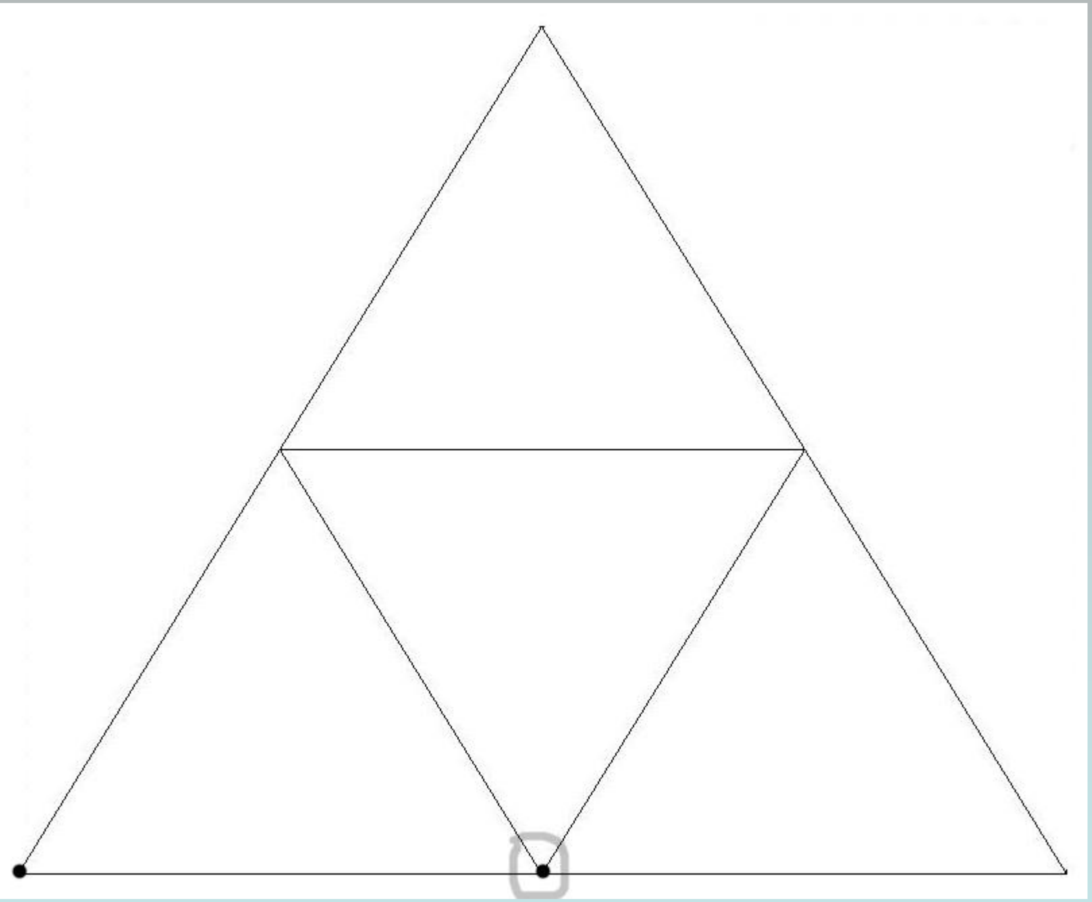


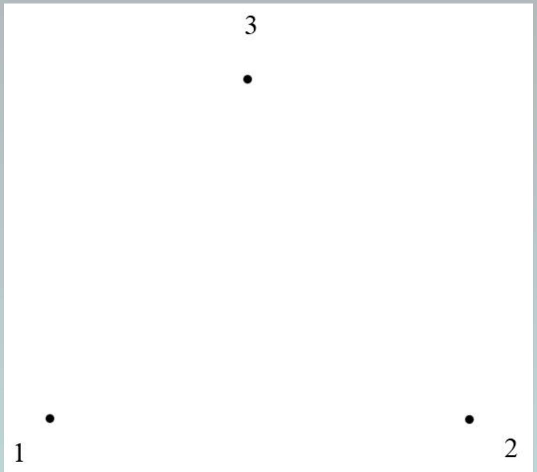
Address of this point is



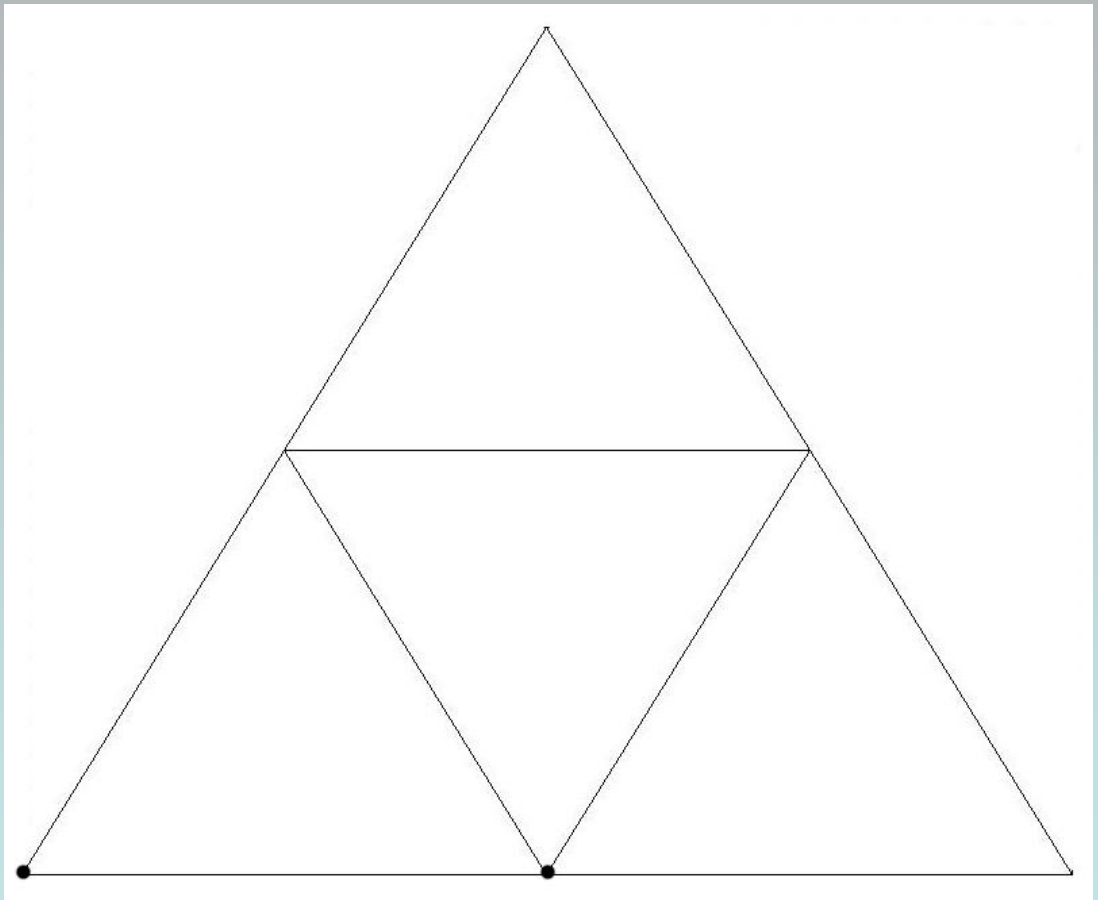
1111111111111111……

Suppose first game number is a 2;



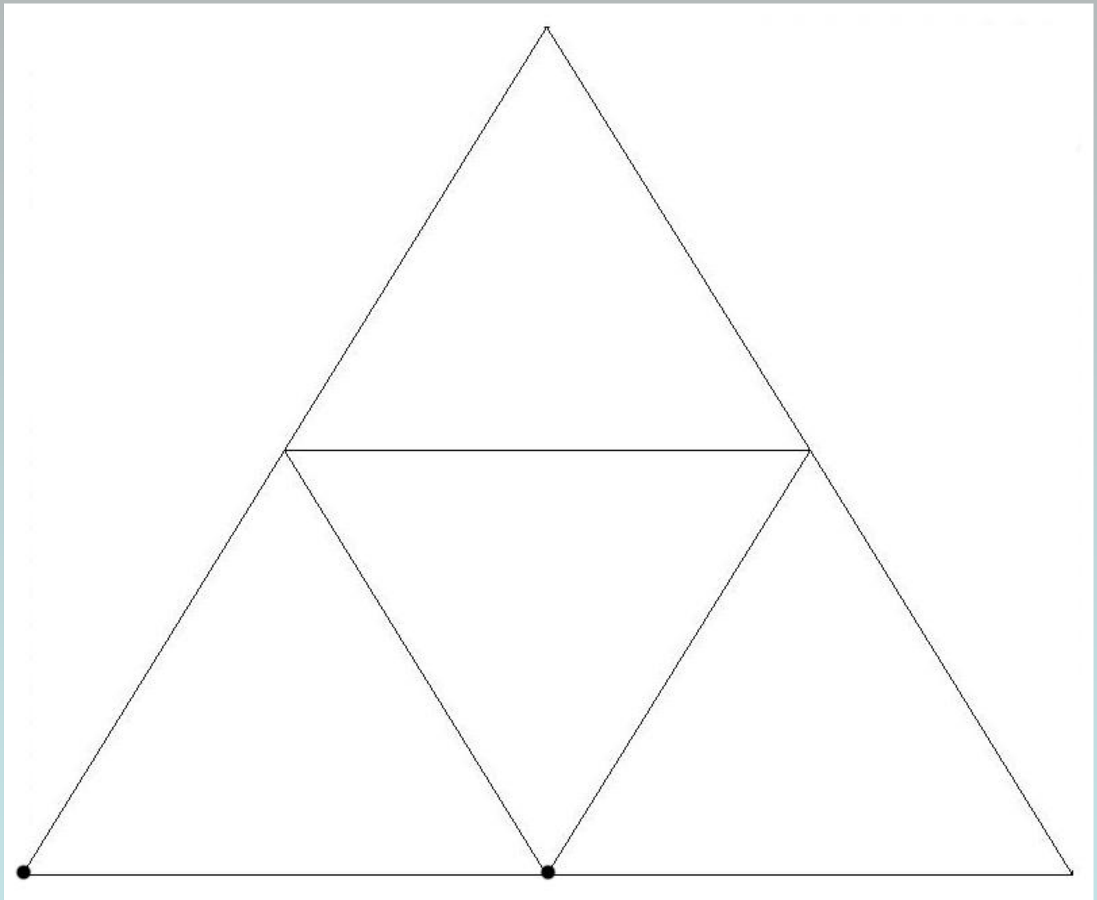


Suppose first game number is a 2;



Address of this game point is

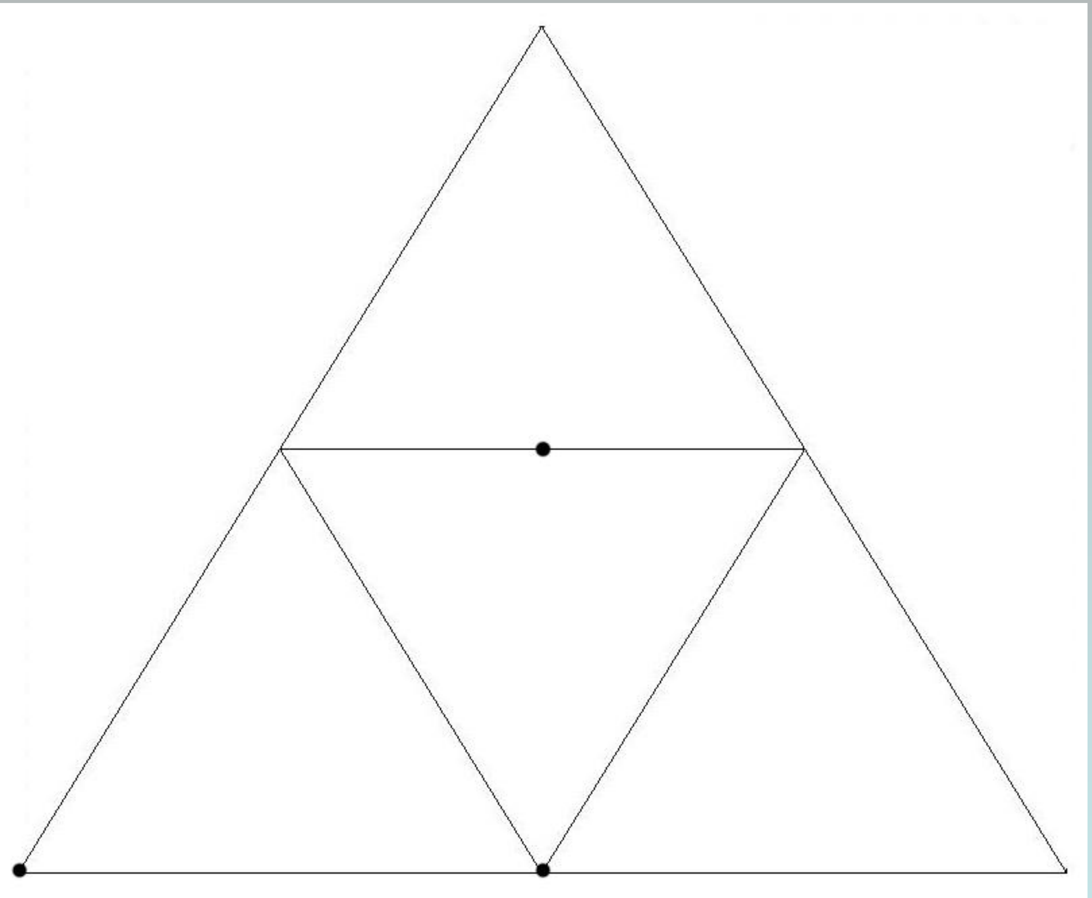
Suppose first game number is a 2;

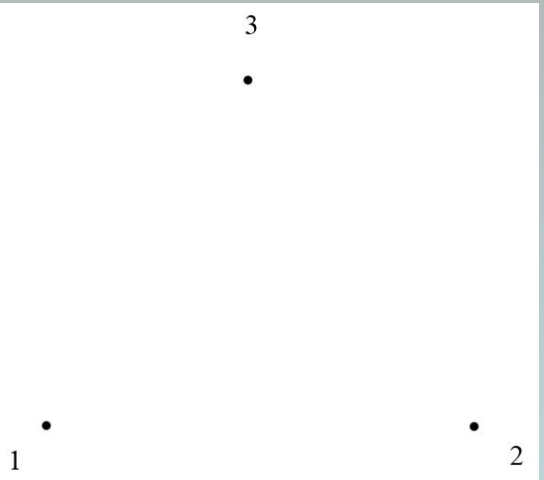


Address of this game point is

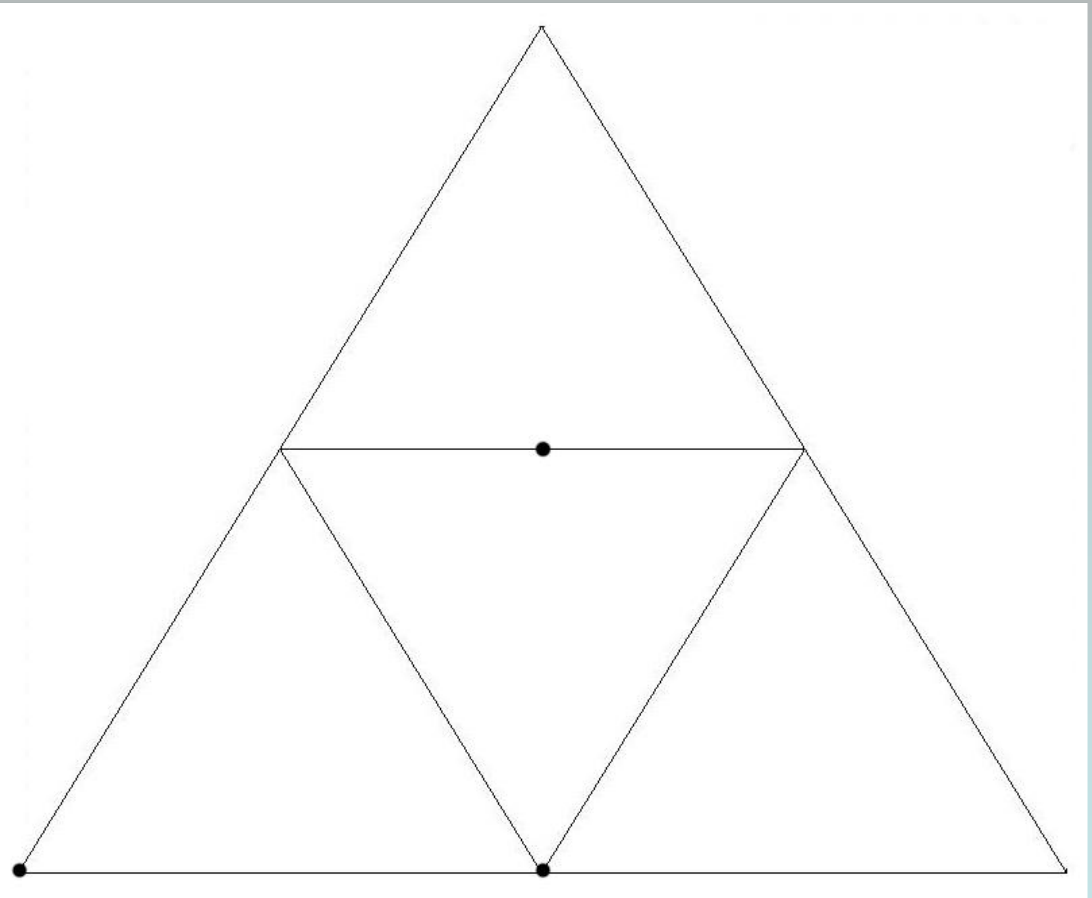
211111111111……..

And if the next game number is a 3;



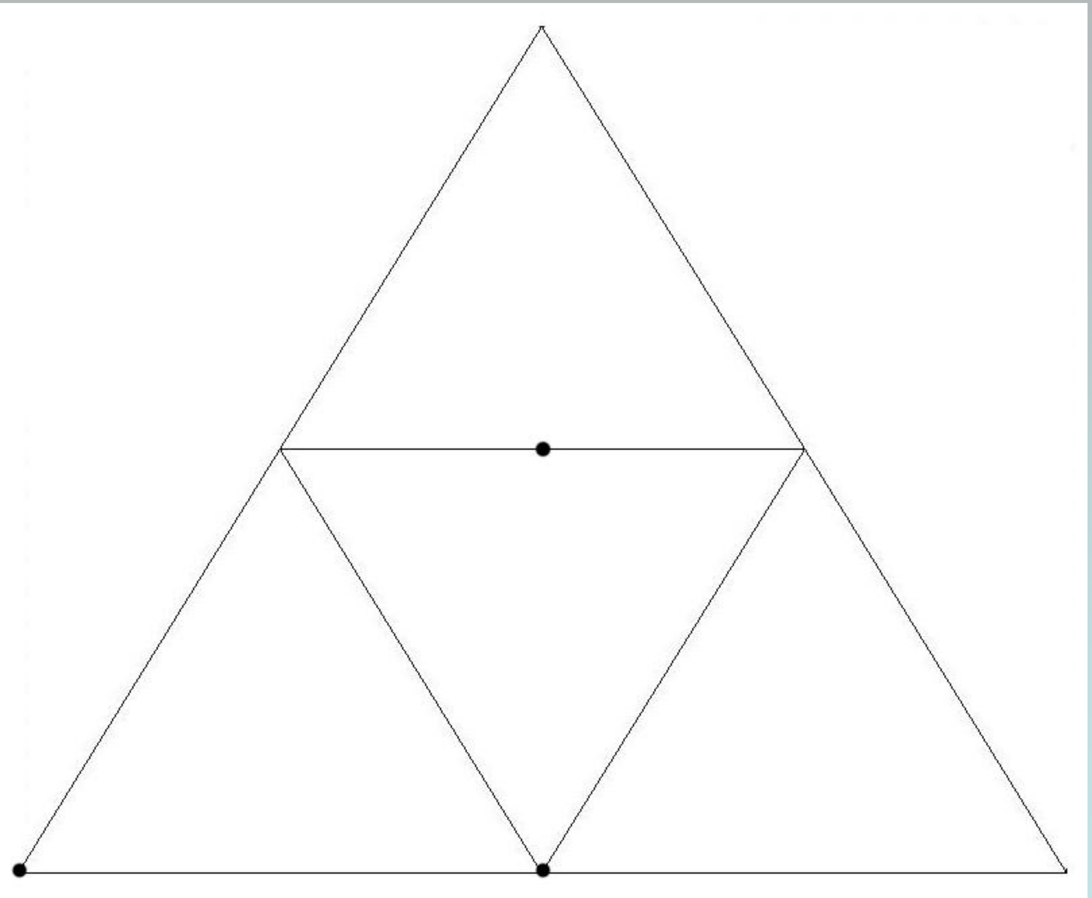


And if the next game number is a 3;



Address of this game point is

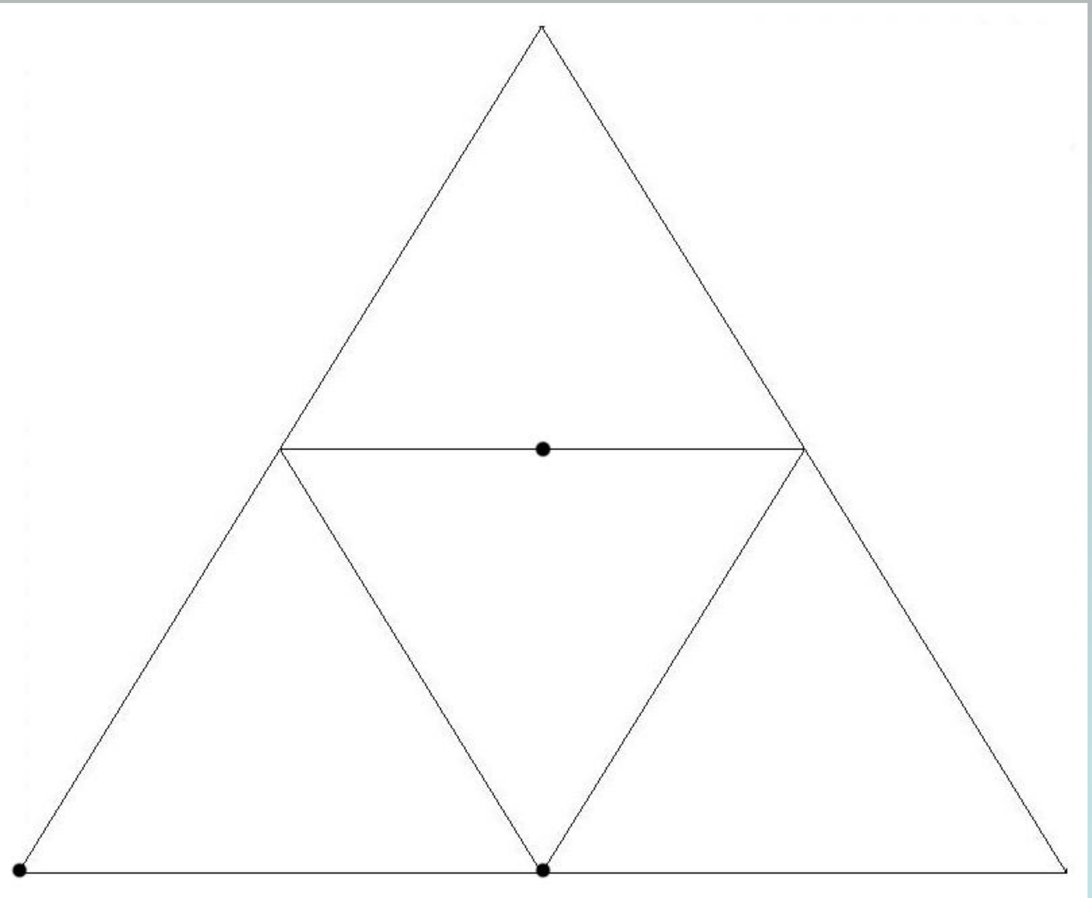
And if the next game number is a 3;



Address of this game point is

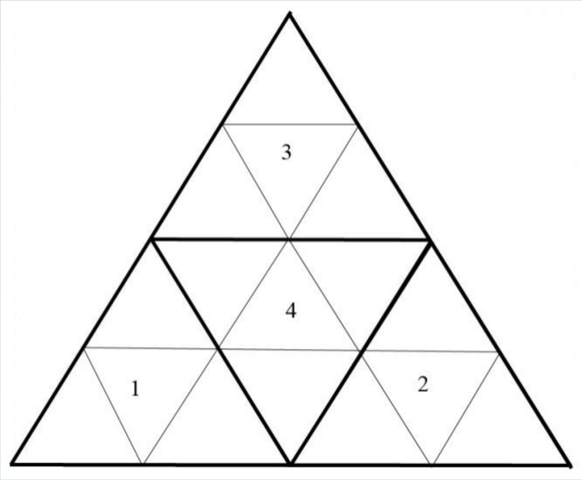
32111111111……..

And if the next game number is a 3;

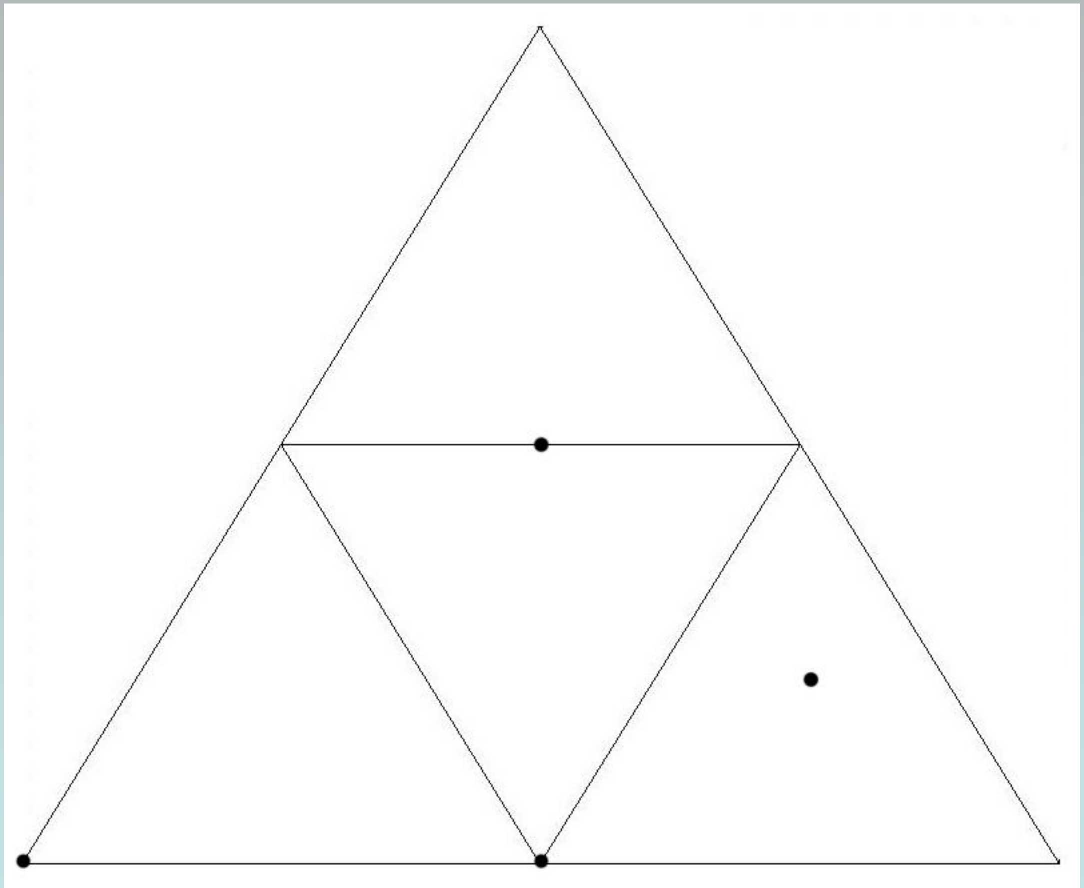


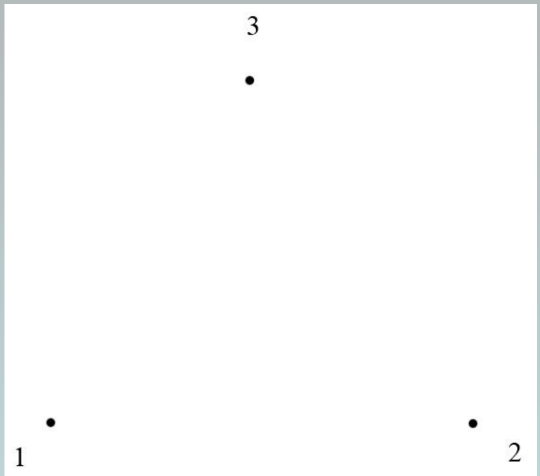
Address of this game point is

32111111111……..

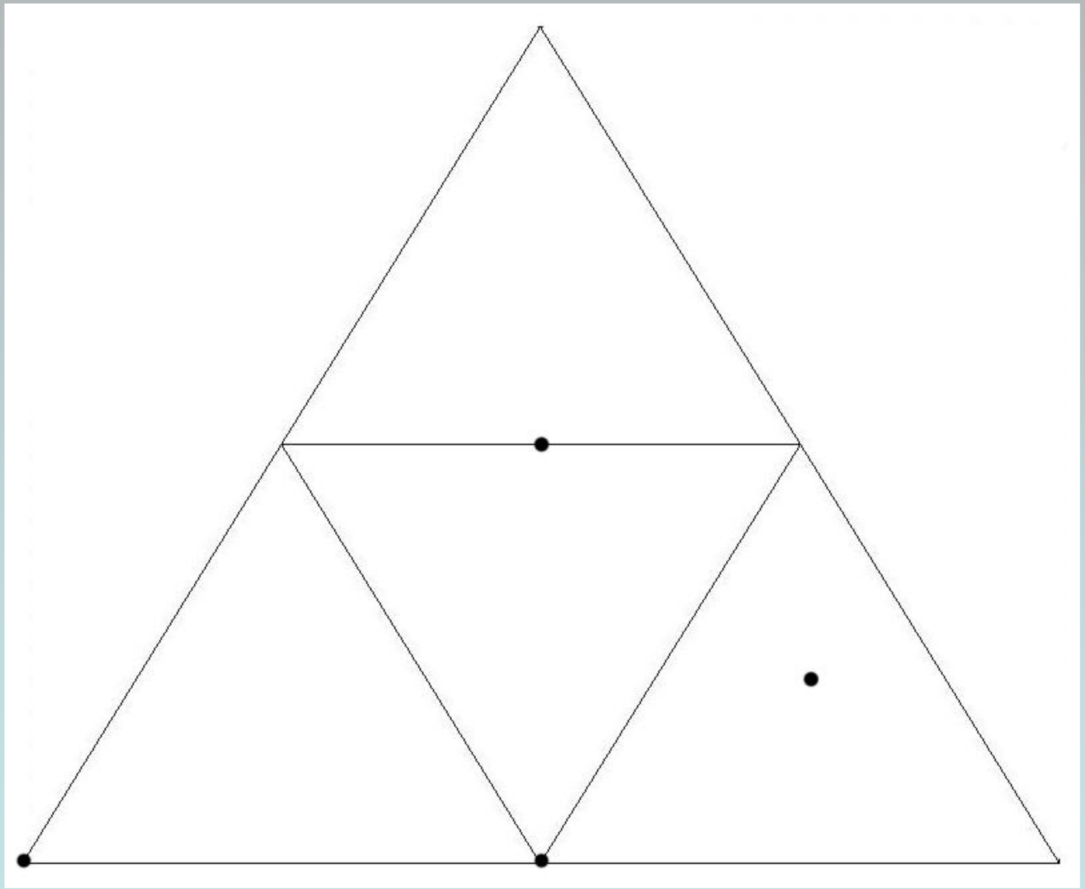


Next game number is a 2;



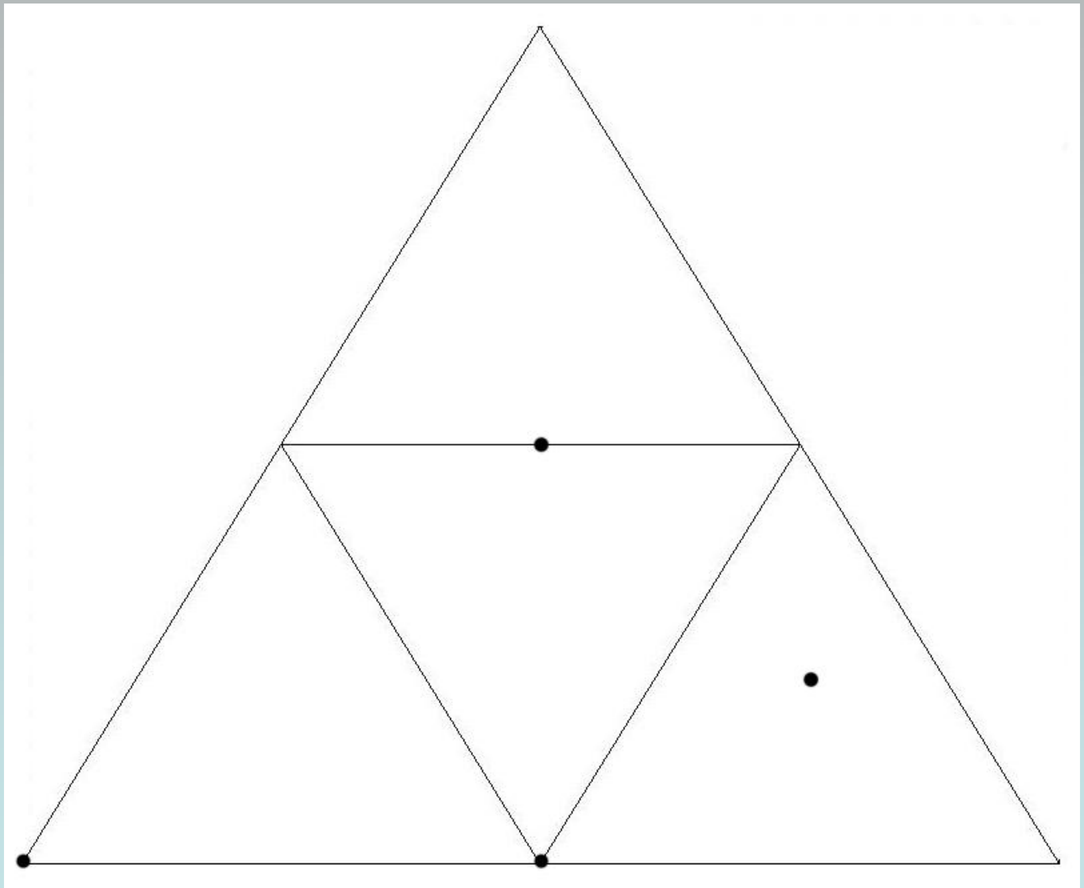


Next game number is a 2;



Address of previous game point is 3211111111…

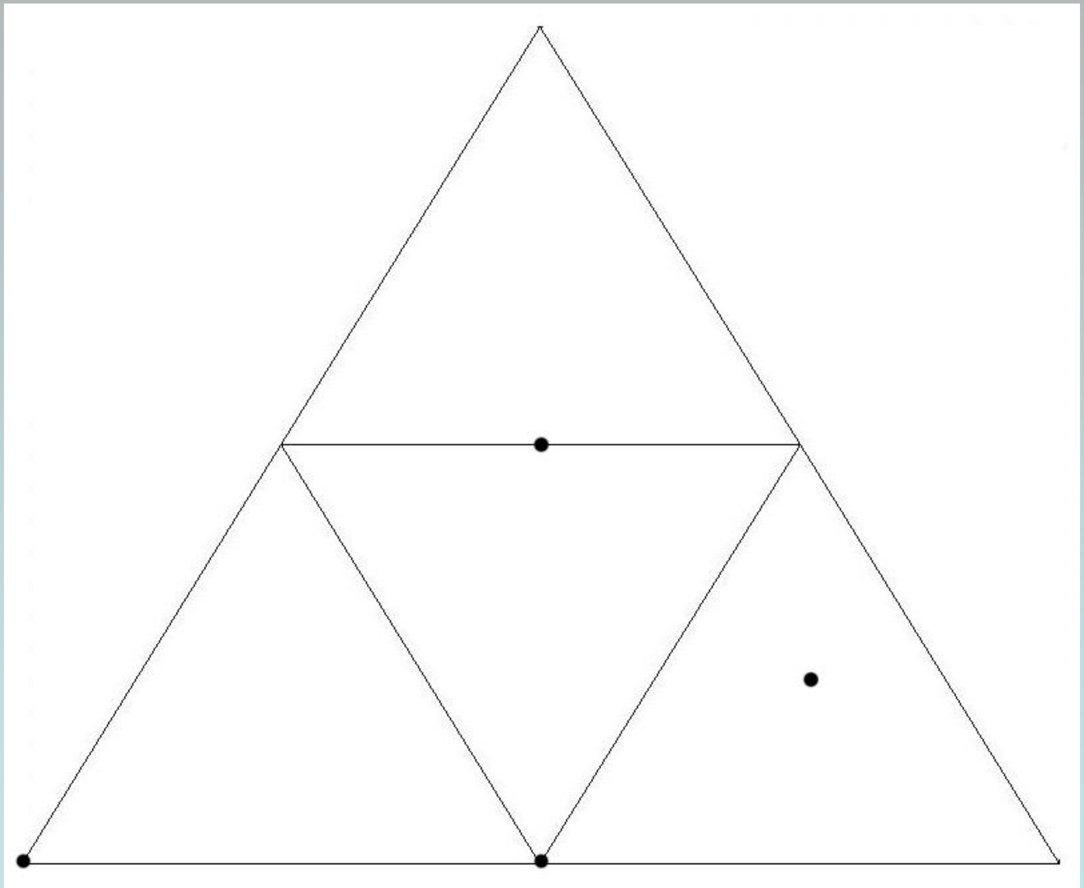
Next game number is a 2;



Address of previous game point is 3211111111…

Address of this game point is

Next game number is a 2;

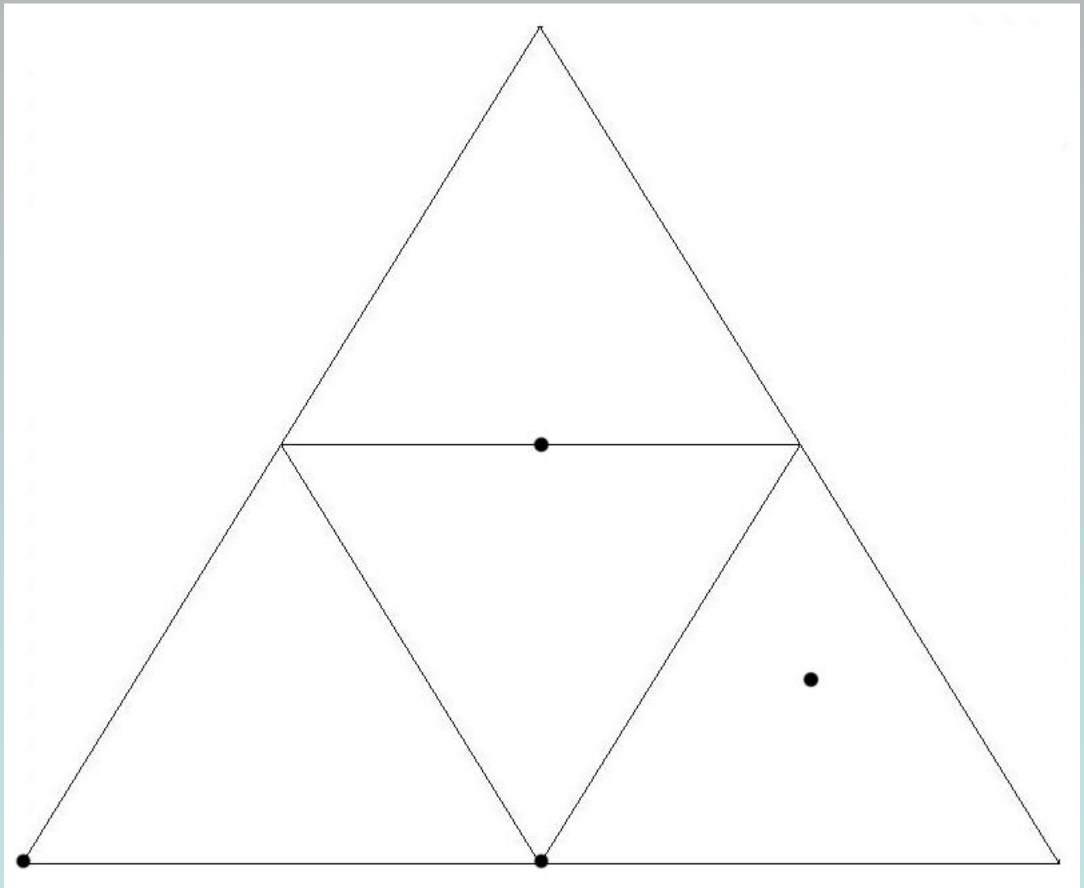


Address of previous game point is 3211111111…

Address of this game point is

23211111111…….

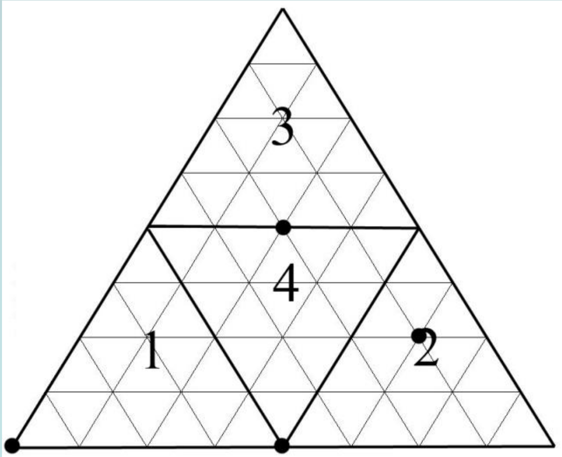
Next game number is a 2;



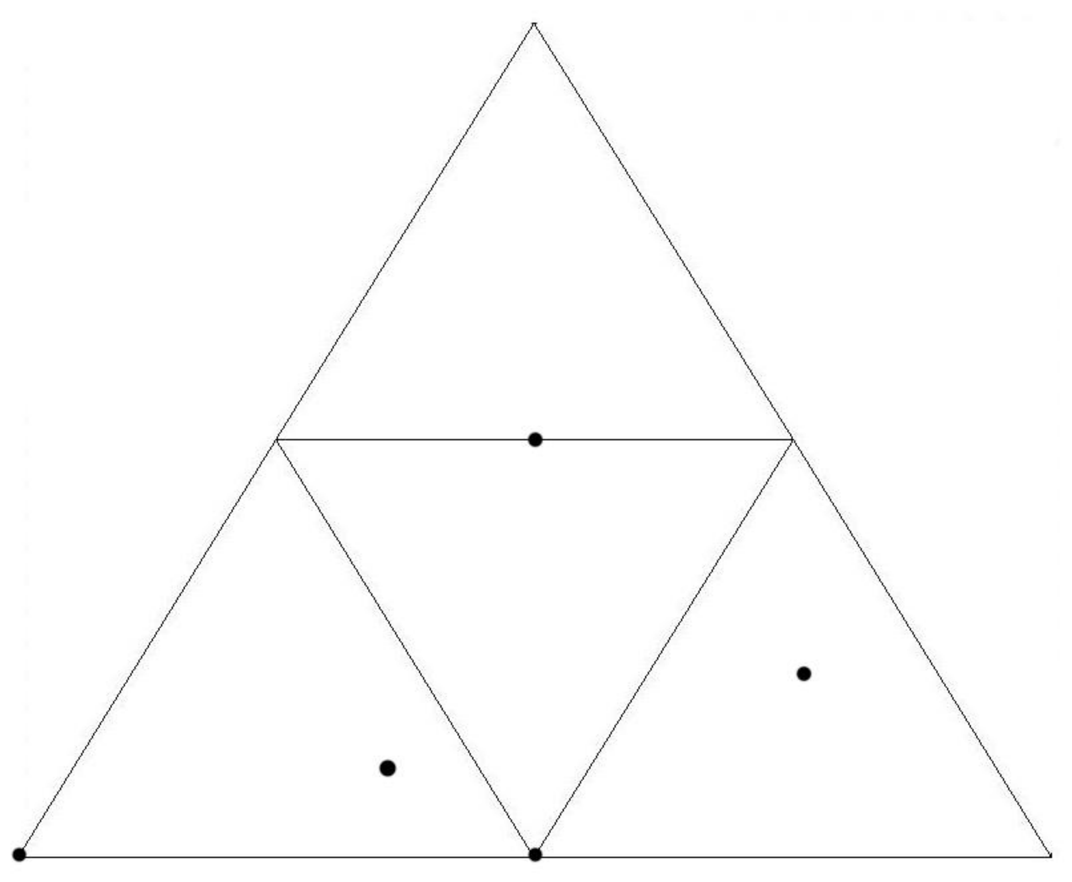
Address of previous game point is 3211111111…

Address of this game point is

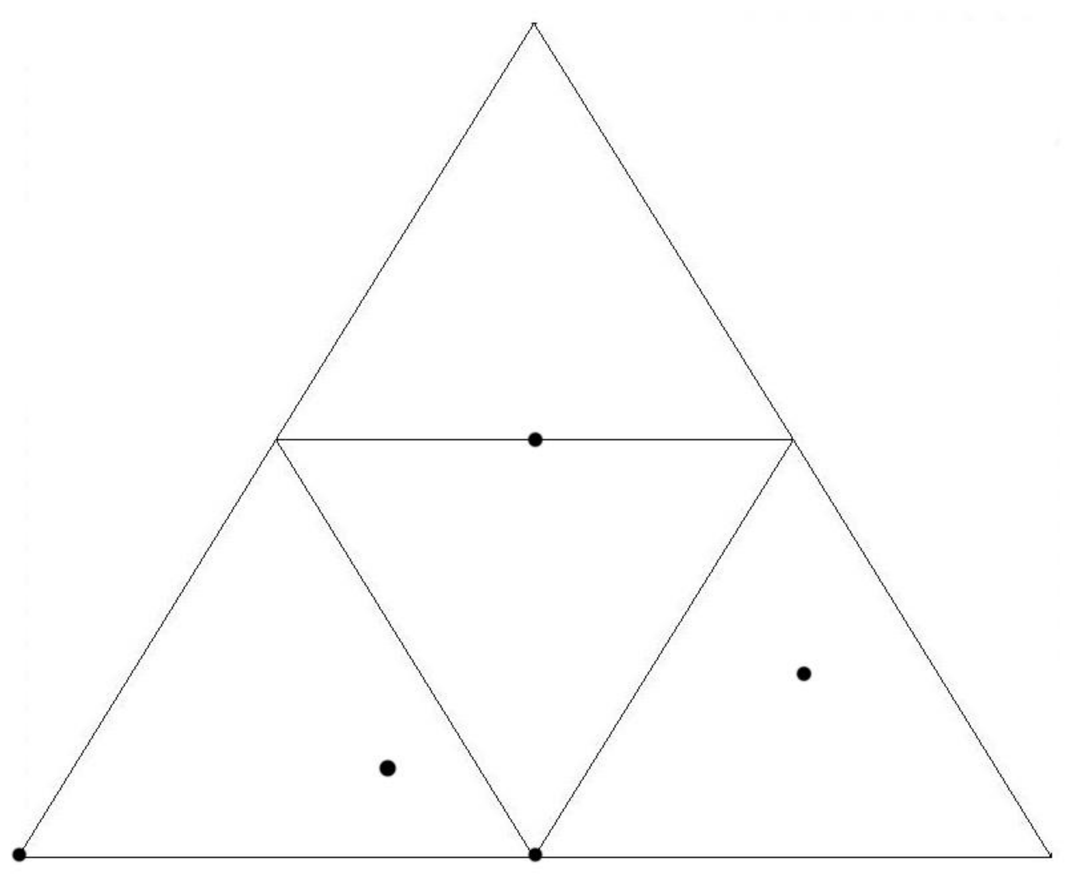
23211111111…….



Next game number is a 1;



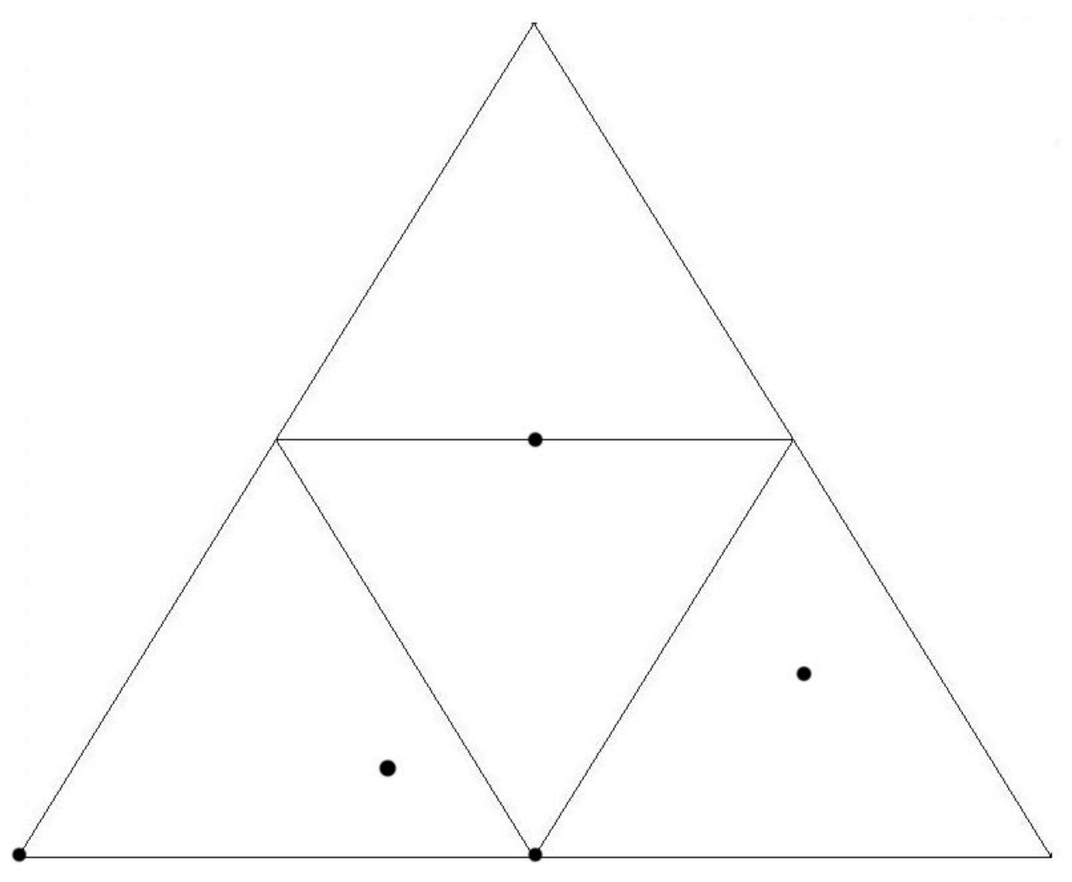
Next game number is a 1;



Address of previous game point is 23211111111…

Address of this game point is

Next game number is a 1;

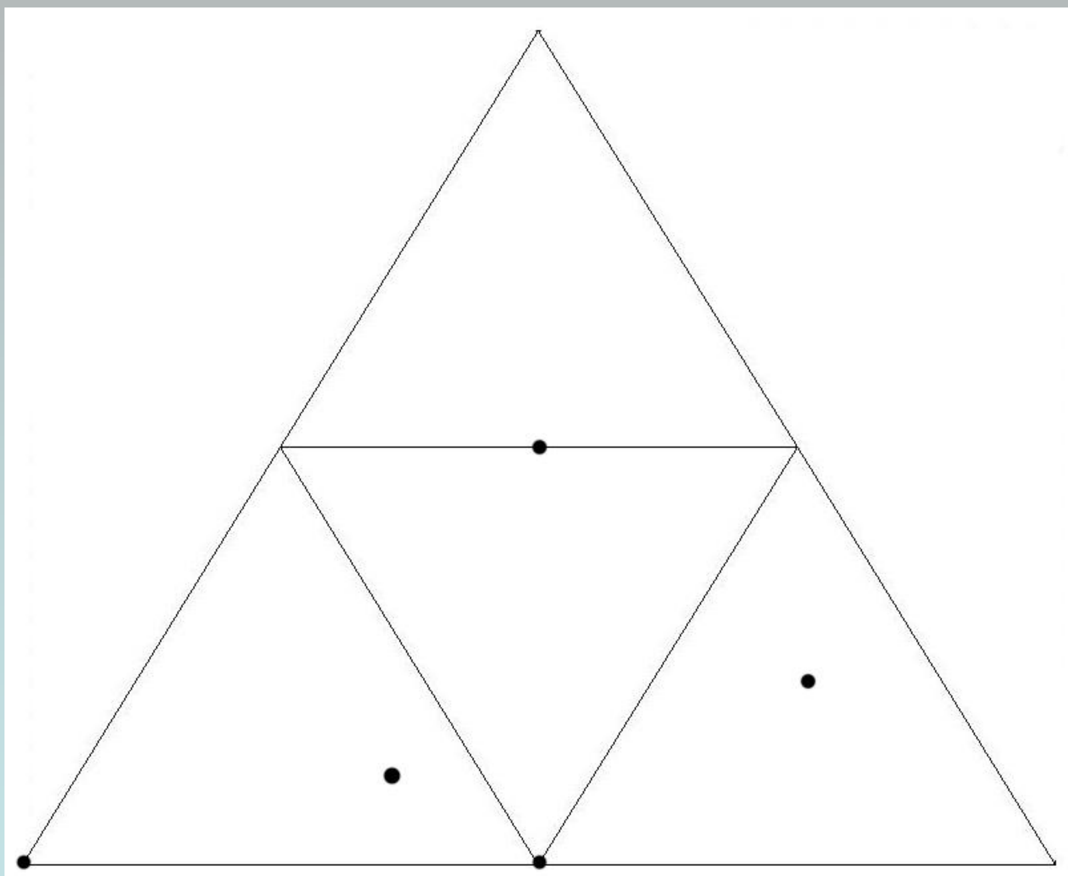


Address of previous game point is 23211111111…

Address of this game point is

123211111111…….

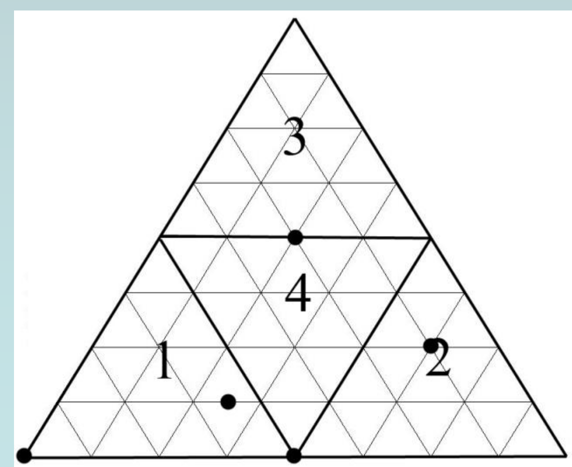
Next game number is a 1;



Address of previous game point is 23211111111…

Address of this game point is

123211111111…….



So, if the game numbers are s1, s2, s3, …, sk,…, the addresses of the game points are

game point 1: s1……...

game point 2: s2s1…….

game point 3: s3s2s1…..

⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮

game point k: sk…….s3s2s1…….

So, if the game numbers are s1, s2, s3, …, sk,…, the addresses of the game points are

game point 1: s1……...

game point 2: s2s1…….

game point 3: s3s2s1…..

⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮

game point k: sk…….s3s2s1…….

(Addresses of points are reversed from sequence)

Which means we can put a game point in <u>every address region</u> of the Sierpinski triangle if the game numbers produce every pattern of 1's, 2's, and 3's.

Here's one sequence of game numbers that produces every pattern of 1's, 2's, and 3's;

Here’s one sequence of game numbers that produces every pattern of 1’s, 2’s, and 3’s;

1 2 3 - first, all patterns of length 1

Here’s one sequence of game numbers that produces every pattern of 1’s, 2’s, and 3’s;

123

11 12 13 21 22 23 31 32 33

- then all patterns of length 2

Here's one sequence of game numbers that produces every pattern of 1's, 2's, and 3's;

123111213212223313233

111 112 113 now all patterns of length 3

121 122 123

131 132 133

211 212 213

221 222 223

231 232 233

311 312 313

321 322 323 …..

12311121321222331323311111211312112

212313113213321121221322122222323…

This sequence will contain every pattern of 1’s, 2’, and 3’s. So if we play the Sierpinski chaos game with this game sequence, the game points will cover all regions of the fractal.

Another sequence that contains all patterns is a *random sequence*

Another sequence that contains all patterns is a *random sequence*

Choose each game number randomly, eg., roll a die;

Another sequence that contains all patterns is a *random sequence*

Choose each game number randomly, eg., roll a die;

if a 1 or 2 comes up, game number is 1

Another sequence that contains all patterns is a *random sequence*

Choose each game number randomly, eg., roll a die;

if a 1 or 2 comes up, game number is 1

if a 3 or 4 comes up, game number is 2

Another sequence that contains all patterns is a *random sequence*

Choose each game number randomly, eg., roll a die;

if a 1 or 2 comes up, game number is 1

if a 3 or 4 comes up, game number is 2

if a 5 or 6 comes up, game number is 3

Another sequence that contains all patterns is a <u>*random sequence*</u>

Choose each game number randomly, eg., roll a die;

if a 1 or 2 comes up, game number is 1

if a 3 or 4 comes up, game number is 2

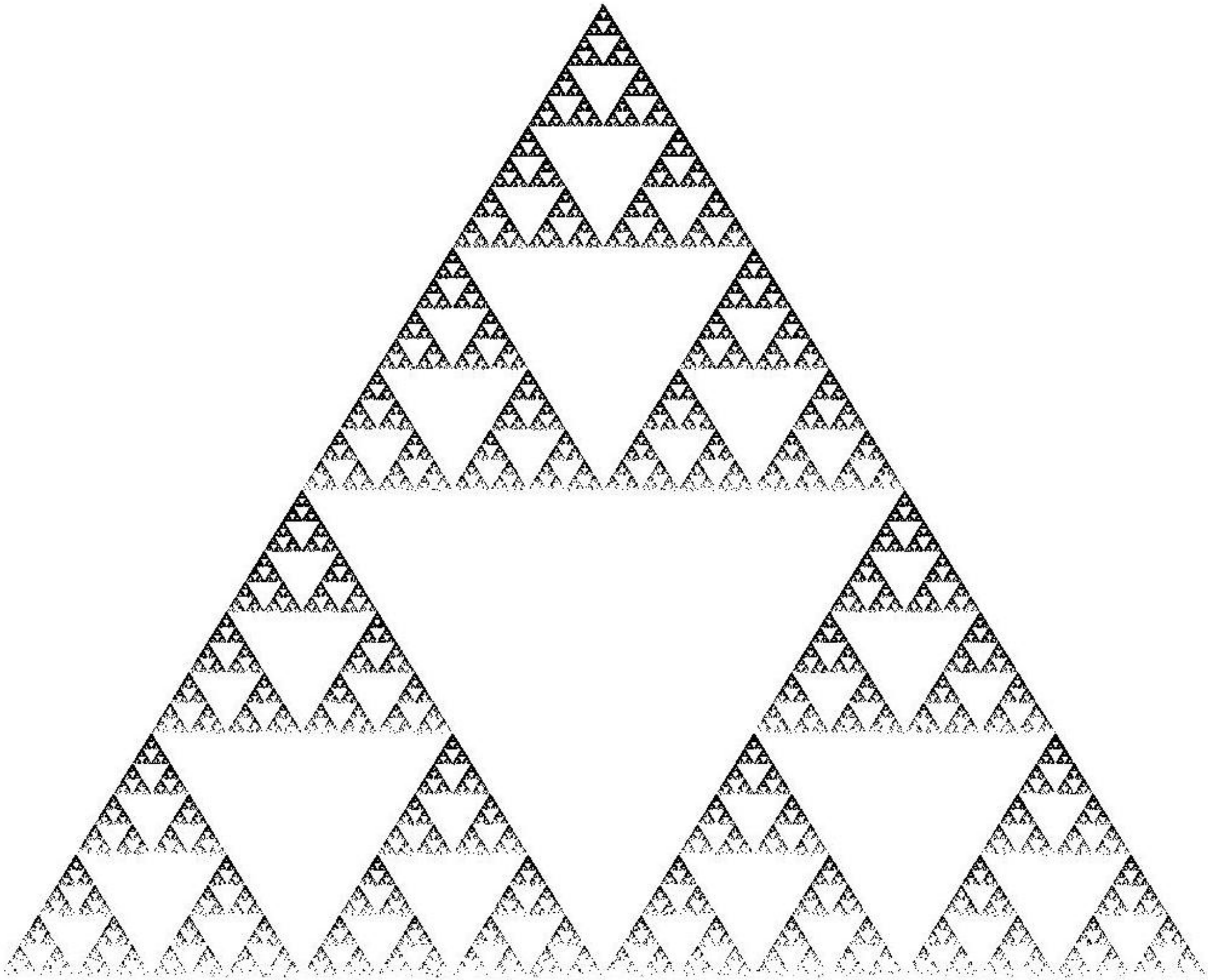
if a 5 or 6 comes up, game number is 3

These game numbers will also draw the fractal.

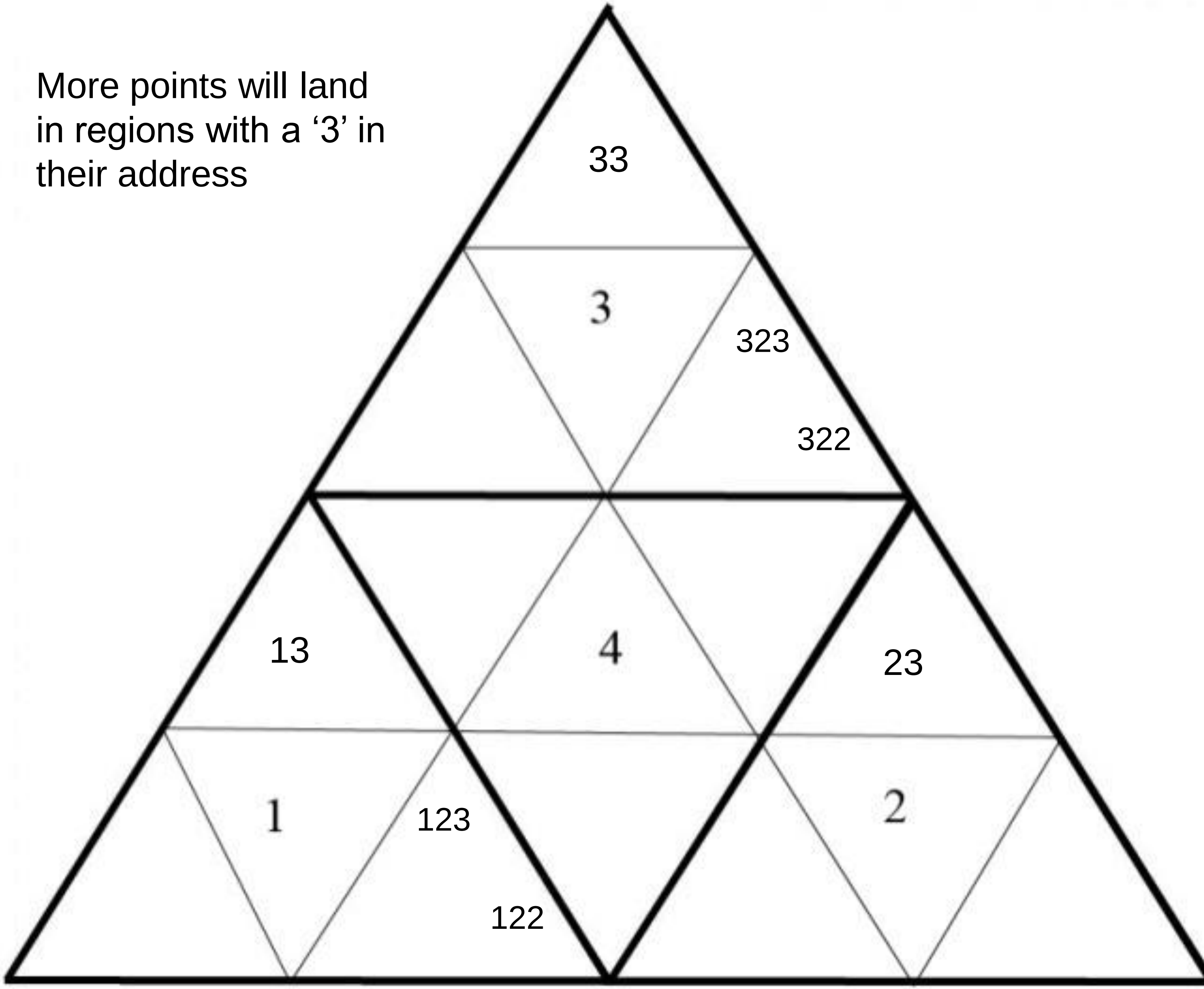
Adjusting probabilities

Suppose we randomly choose 1, 2, 3, but we choose 3 2/3 of the time and 1 and 2 1/6 of the time each (roll a die: if 1 comes up choose 1, if 2 comes up choose 2, if 3,4,5 or 6 come up choose 3);

$$p_1 = \frac{1}{6}, \quad p_2 = \frac{1}{6}, \quad p_3 = \frac{2}{3}$$



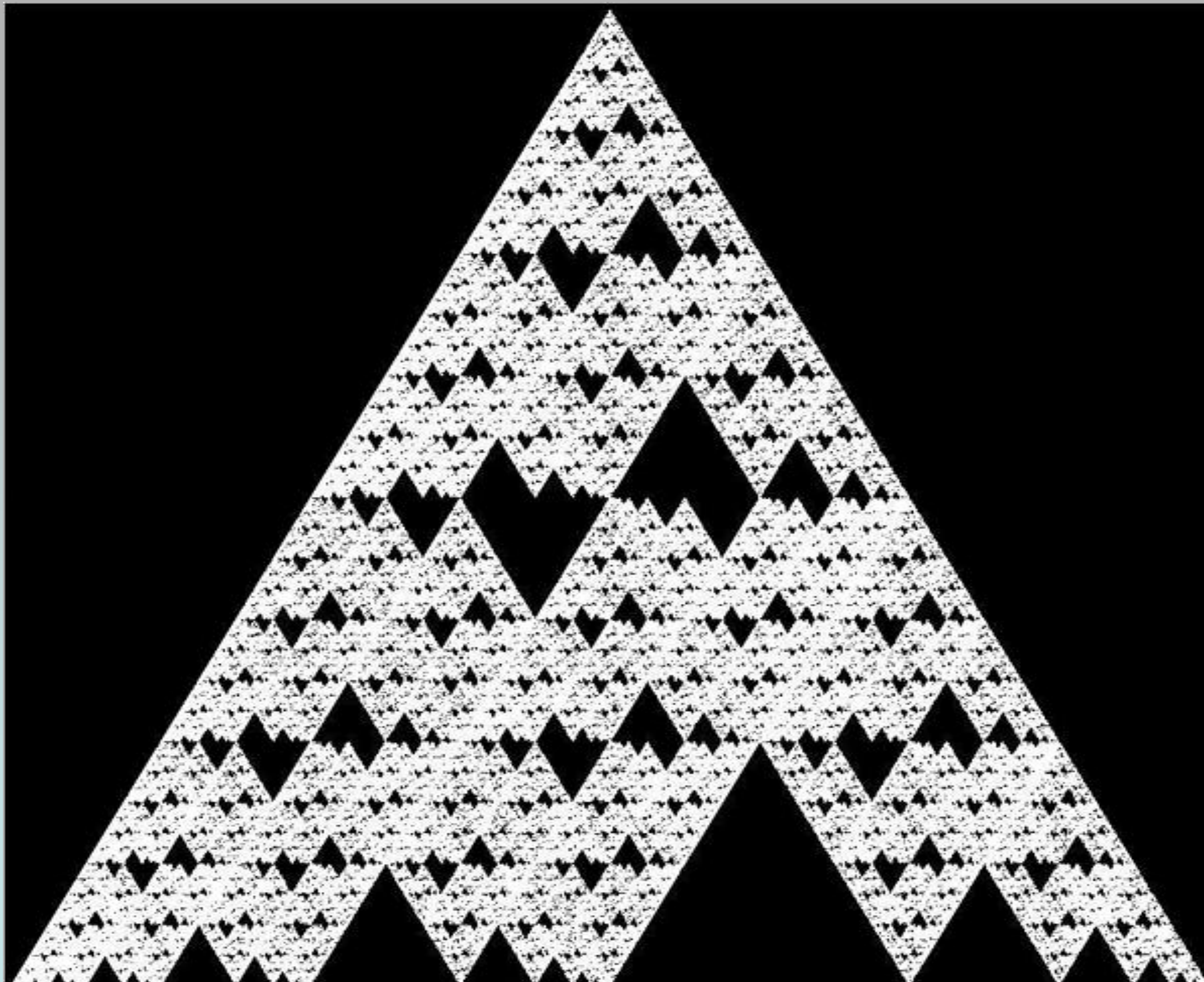
More points will land in regions with a ‘3’ in their address



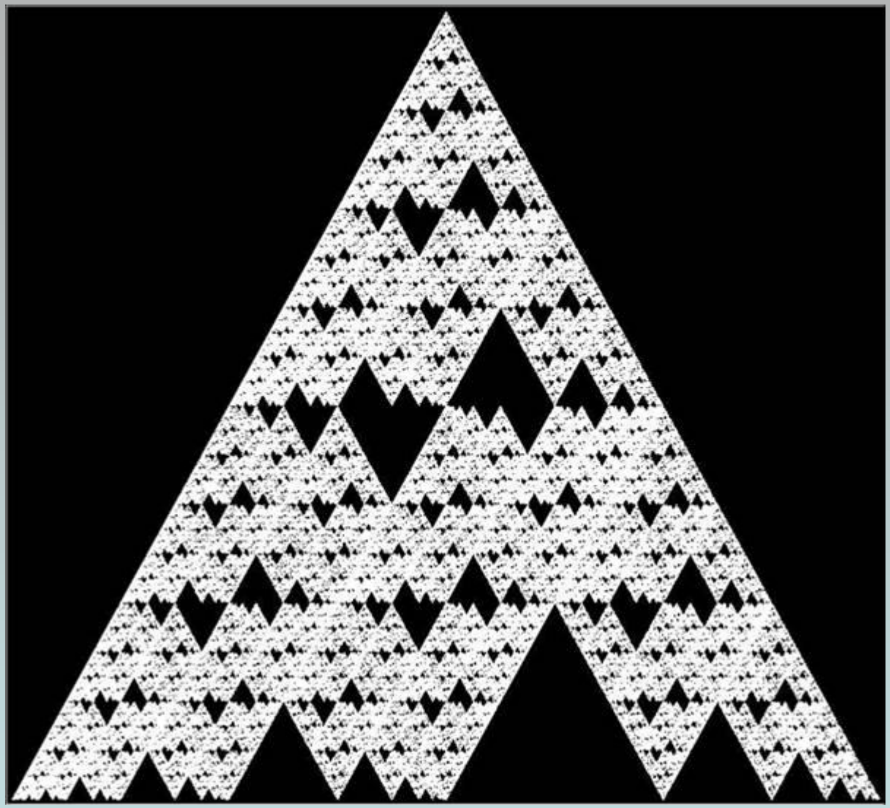
Pseudo Fractals

Full Triangle Pseudo Fractal

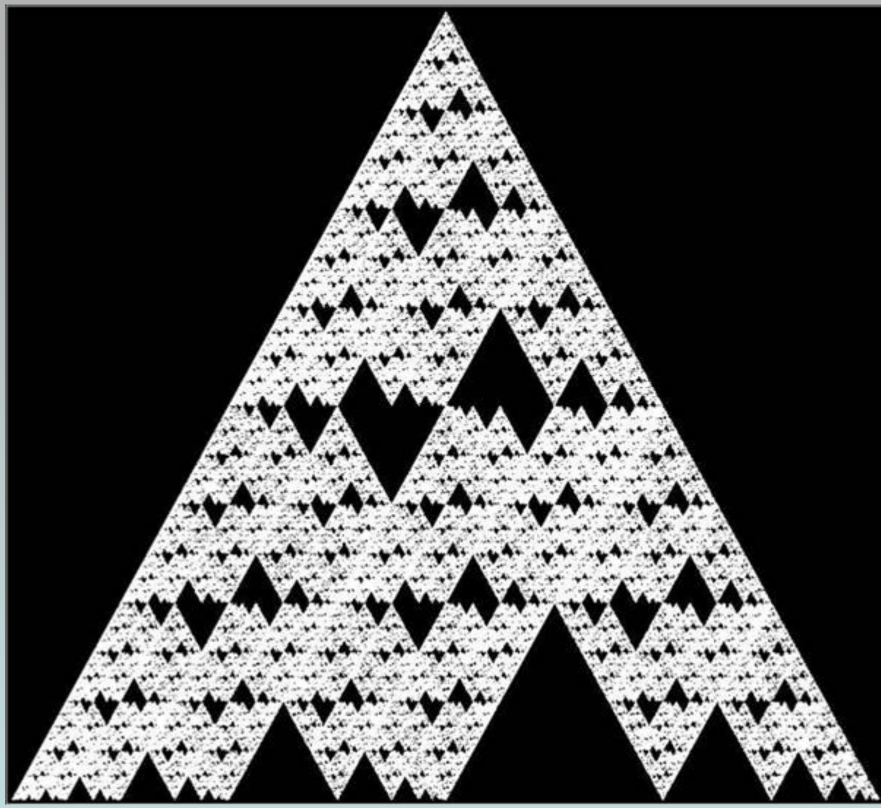
- Same set up as for the Full Triangle Fractal, but generate the sequence of random numbers first.
- From this sequence remove all occurrences of '12'
- Now follow the game rules for the Full Triangle

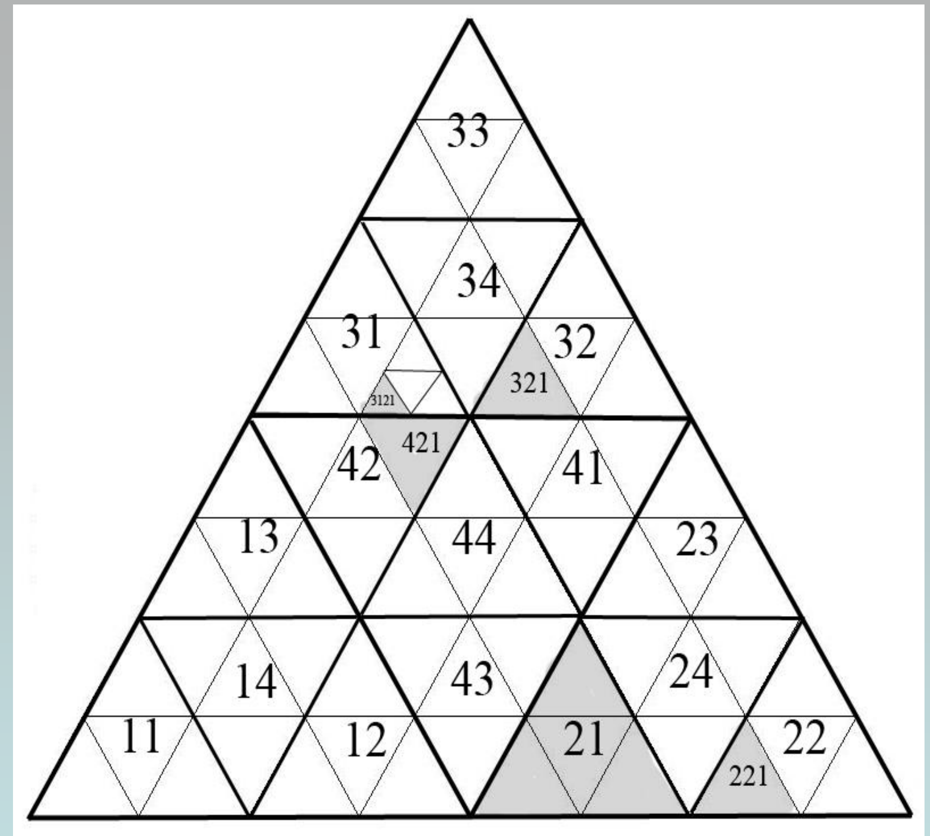


Remove all 12’s from game numbers…..



Remove all 12's from game numbers…..

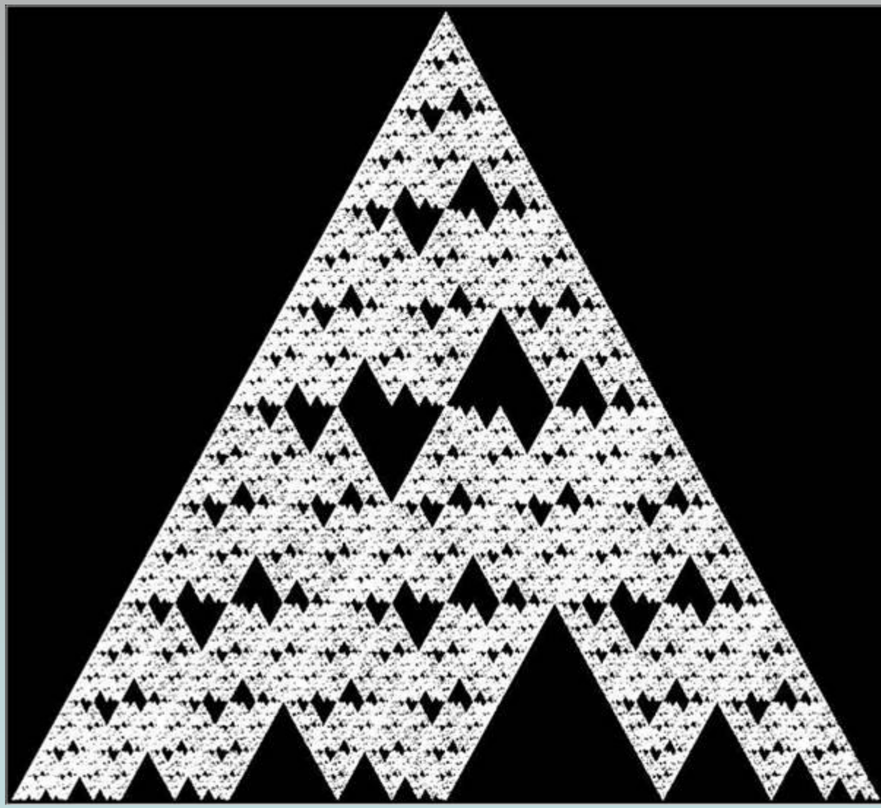


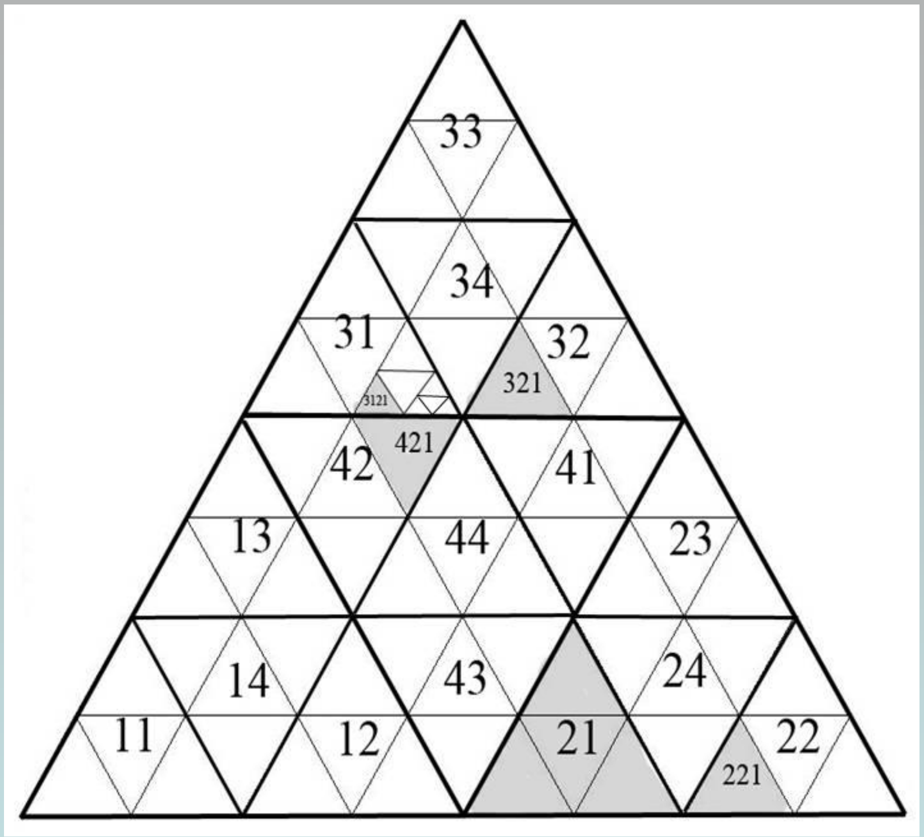


Remember, addresses of game points are the <u>reverse</u> of the game numbers.

So here no game points land in areas whose address contains a 21

Remove all 12's from game numbers…..

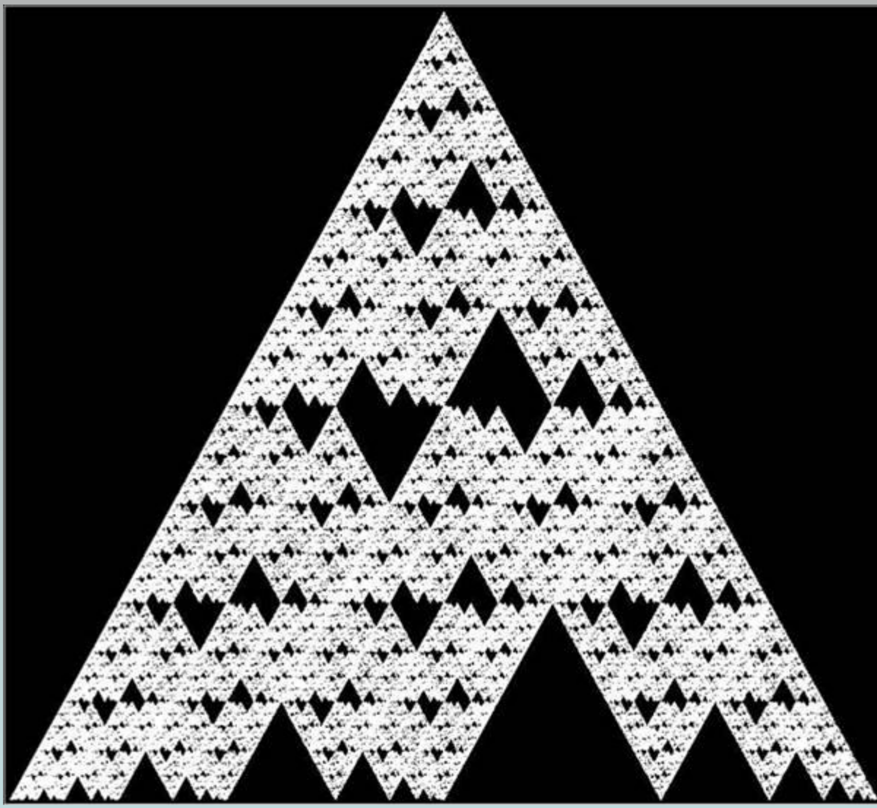


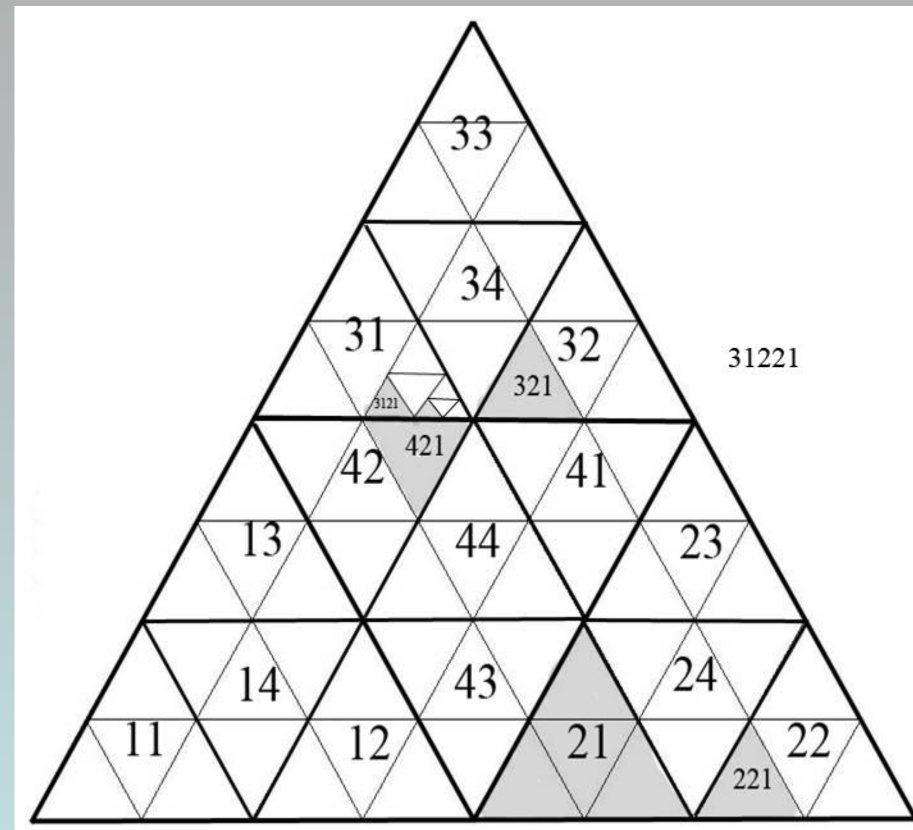


Remember, addresses of game points are the <u>reverse</u> of the game numbers.

So here no game points land in areas whose address contains a 21

Remove all 12's from game numbers…..

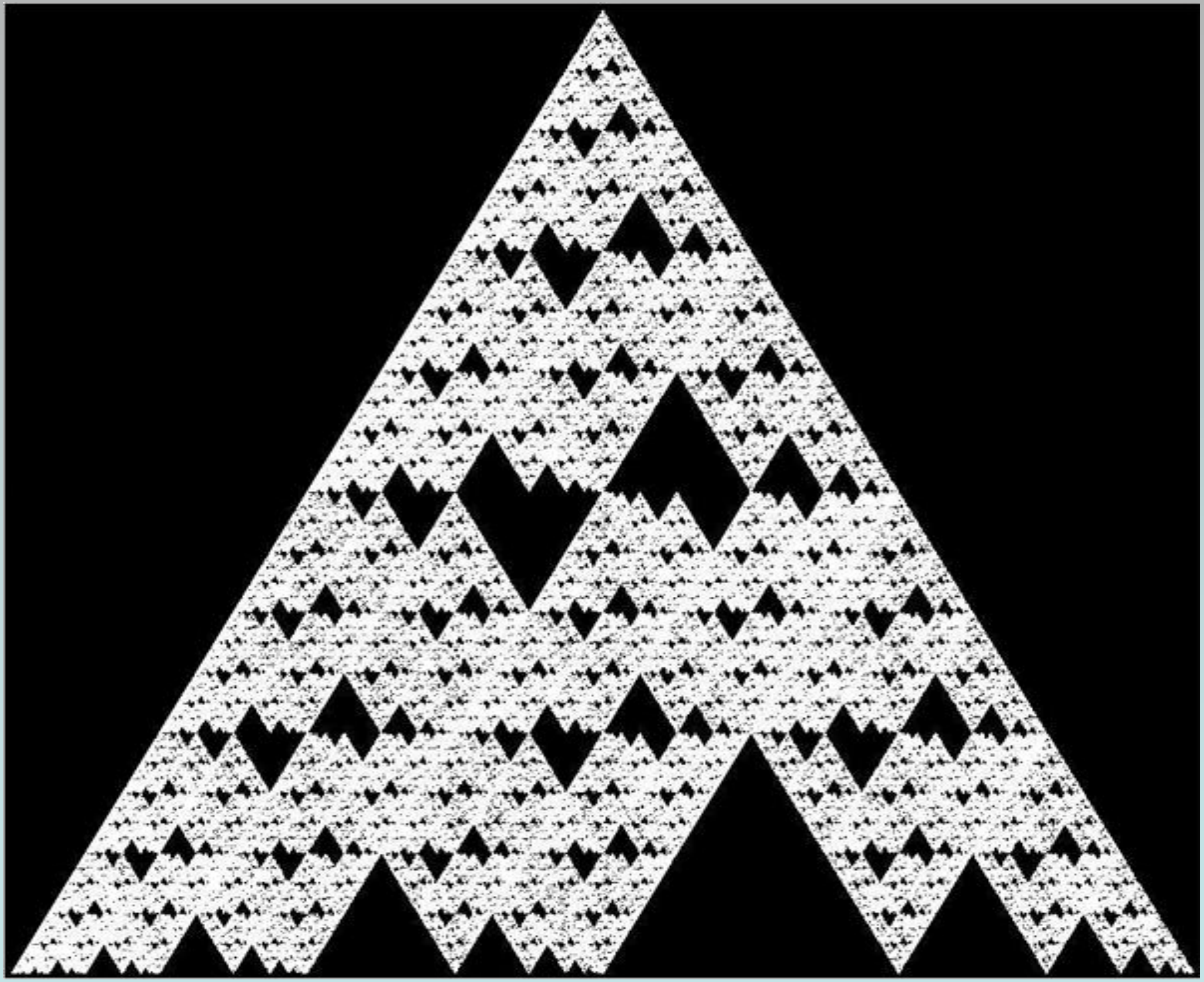


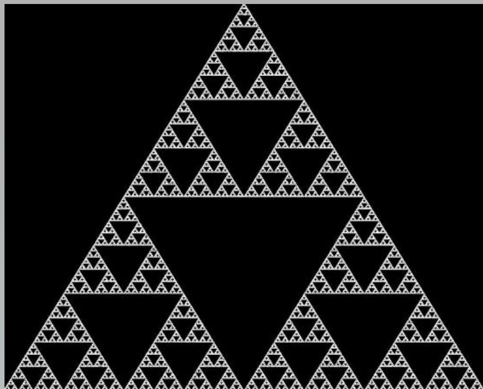
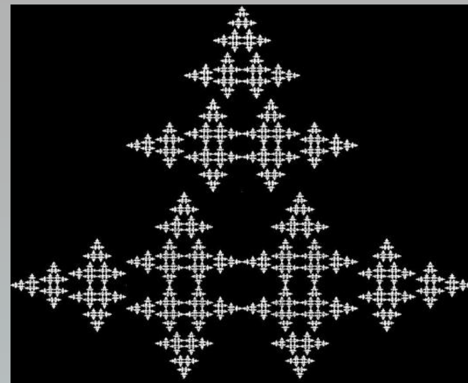
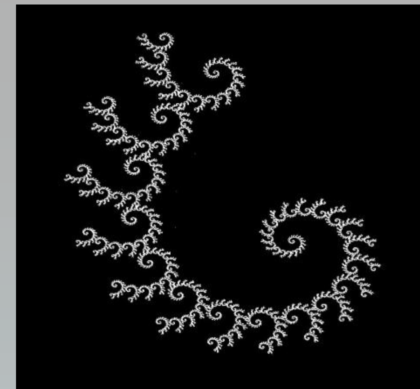


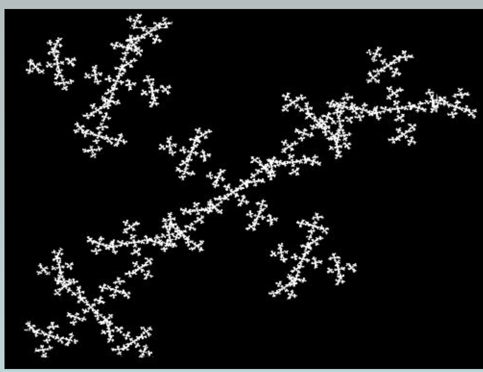
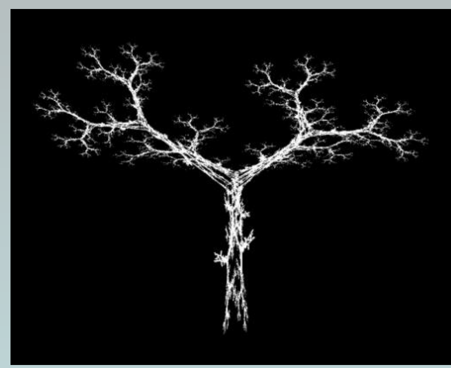
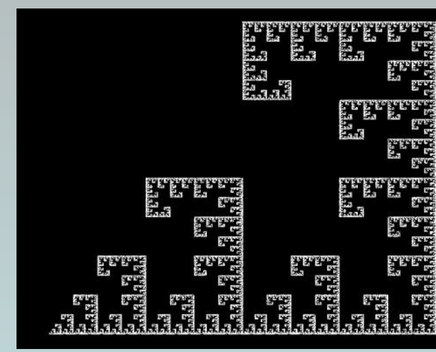
Remember, addresses of game points are the <u>reverse</u> of the game numbers.

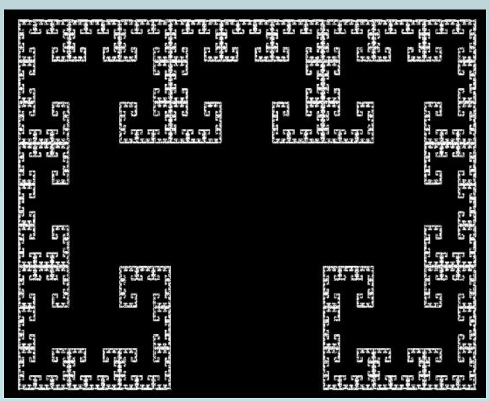
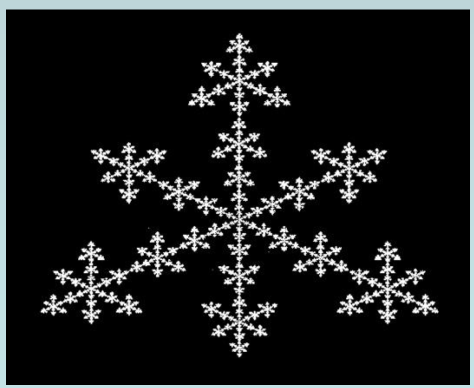
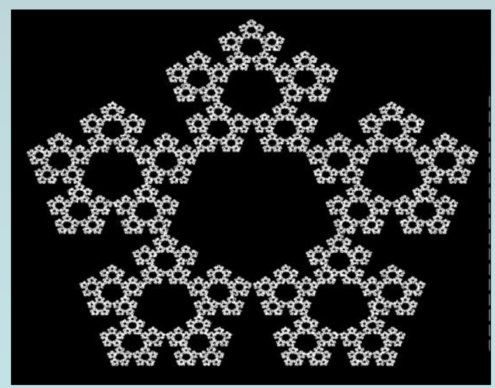
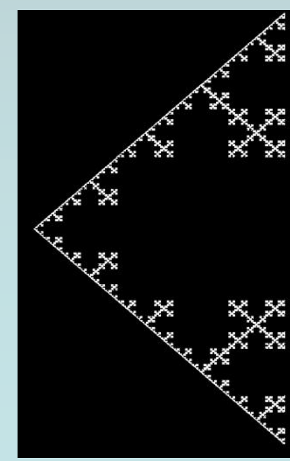
So here no game points land in areas whose address contains a 21

Note; this is <u>not</u> a fractal (is not self-similar)



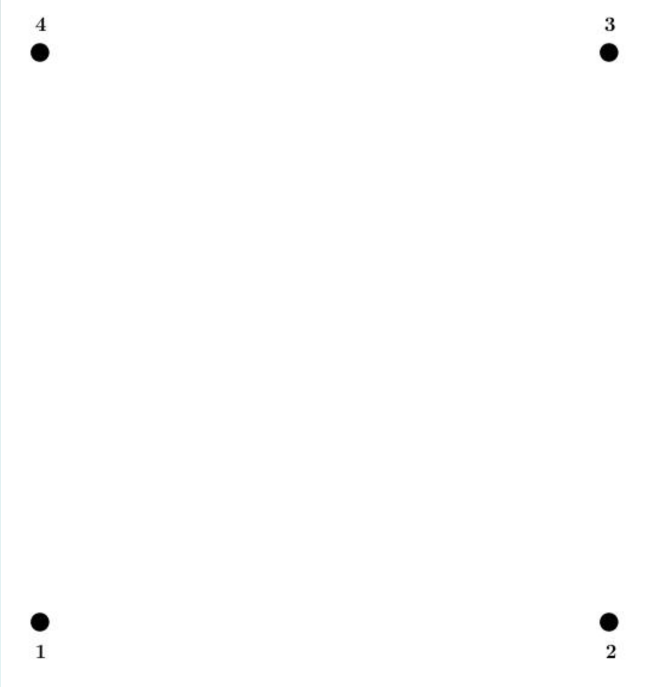





Back to adjusting probabilities

Back to adjusting probabilities

Full Square

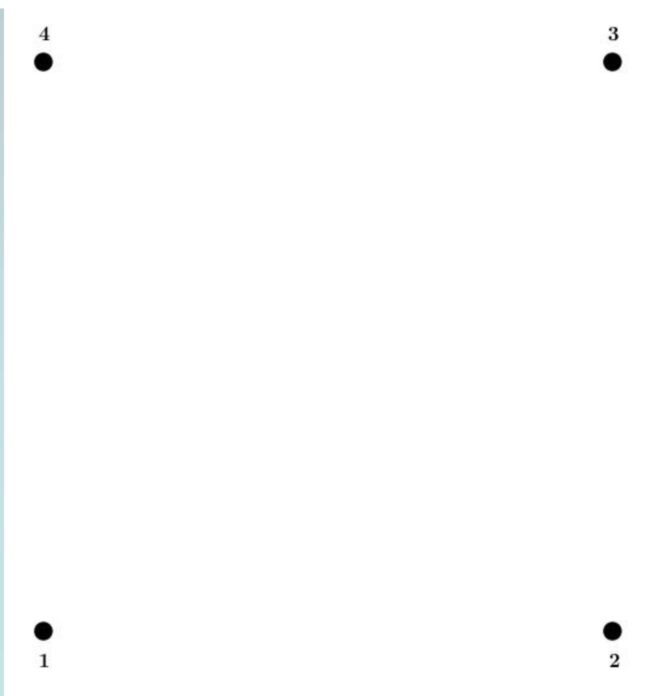
- four pins at the corners of a square
- choose random number $\mathbf{s_i}$ from $\{1, 2, 3, 4\}$
- move 1/2 distance to pin labelled $\mathbf{s_i}$



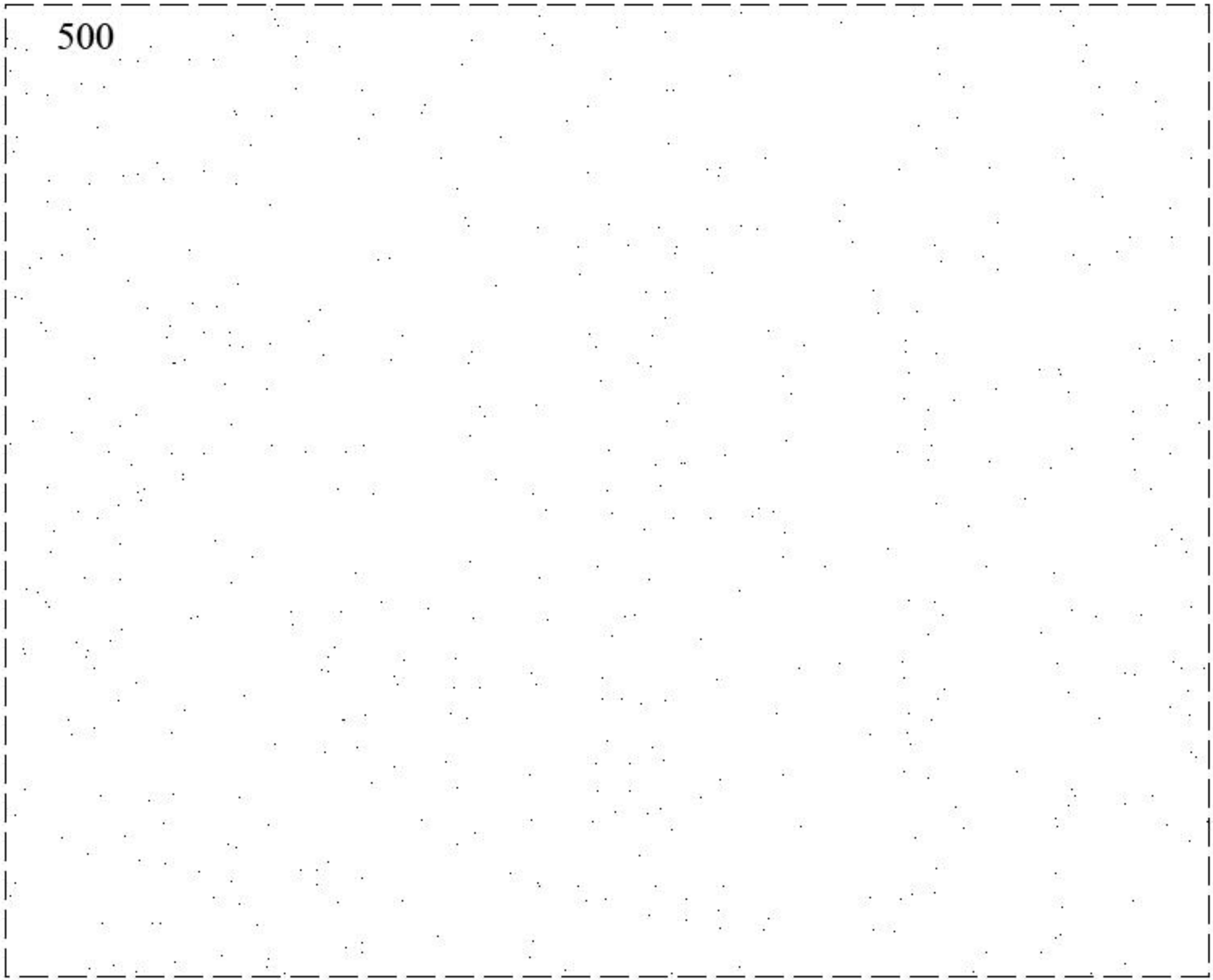
Back to adjusting probabilities

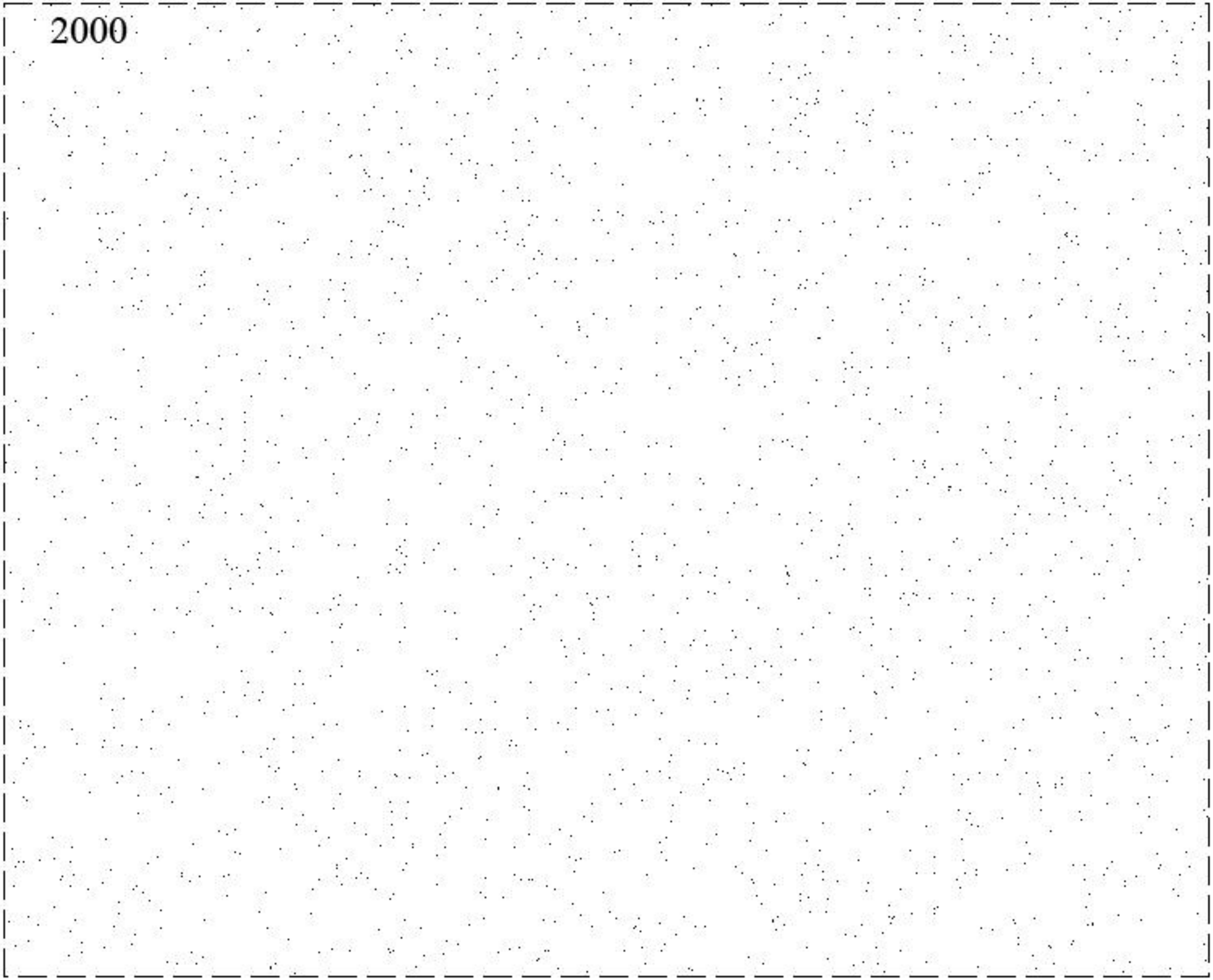
Full Square

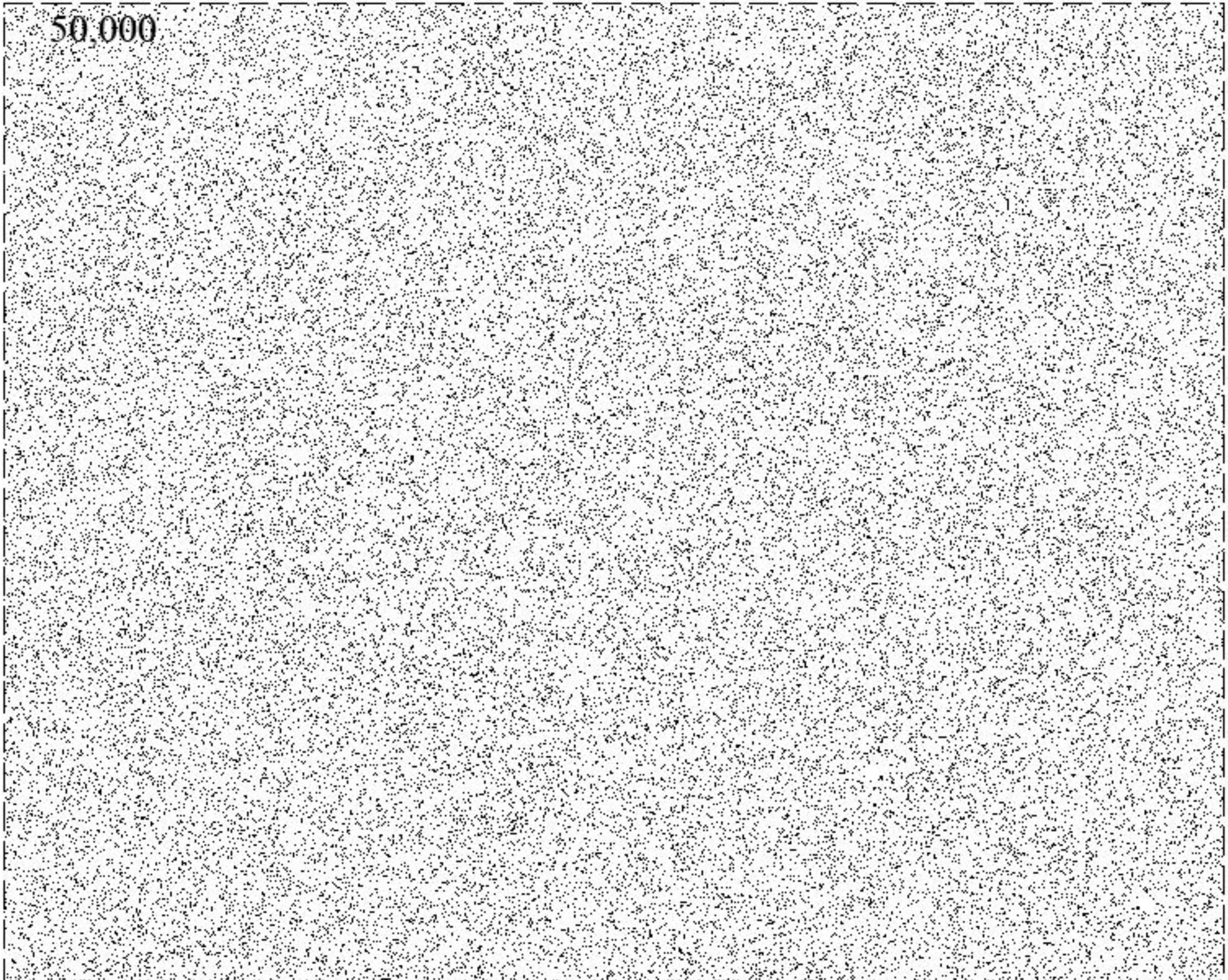
- four pins at the corners of a square
- choose random number $\mathbf{s_i}$ from $\{1, 2, 3, 4\}$
- move 1/2 distance to pin labelled $\mathbf{s_i}$



Equal probabilities


500

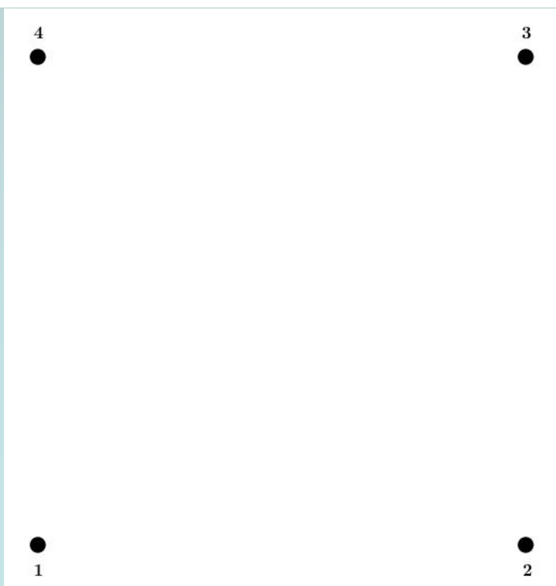

2000

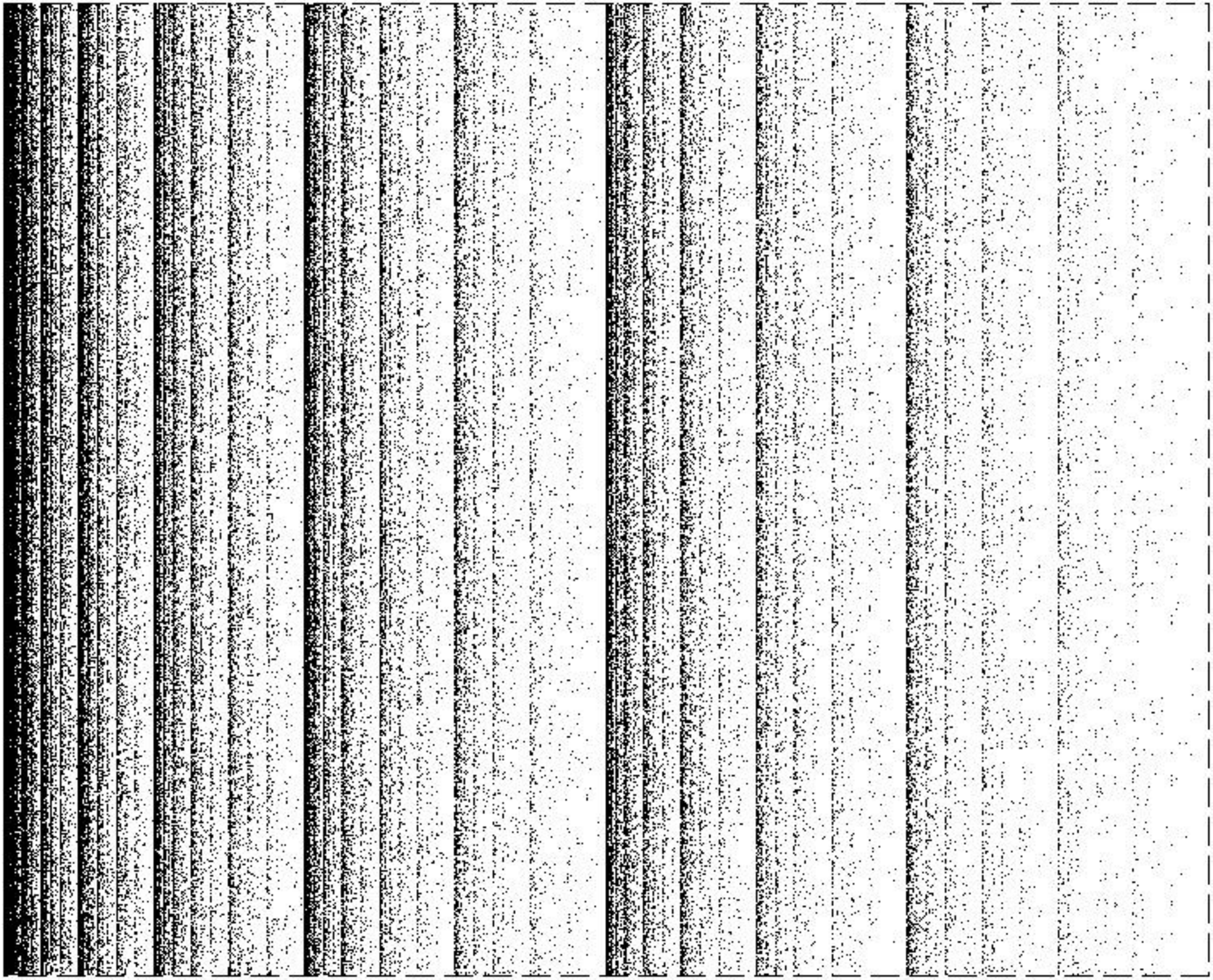

50,000

Unequal probabilities;

Full Square (again)

- four pins at the corners of a square
- choose random number $\mathbf{s_i}$ from $\{1, 2, 3, 4\}$ **but this time choose 1 and 4 40% of the time each, and 2 and 3 10% of the time each**
- move $1/2$ distance to pin labelled $\mathbf{s_i}$

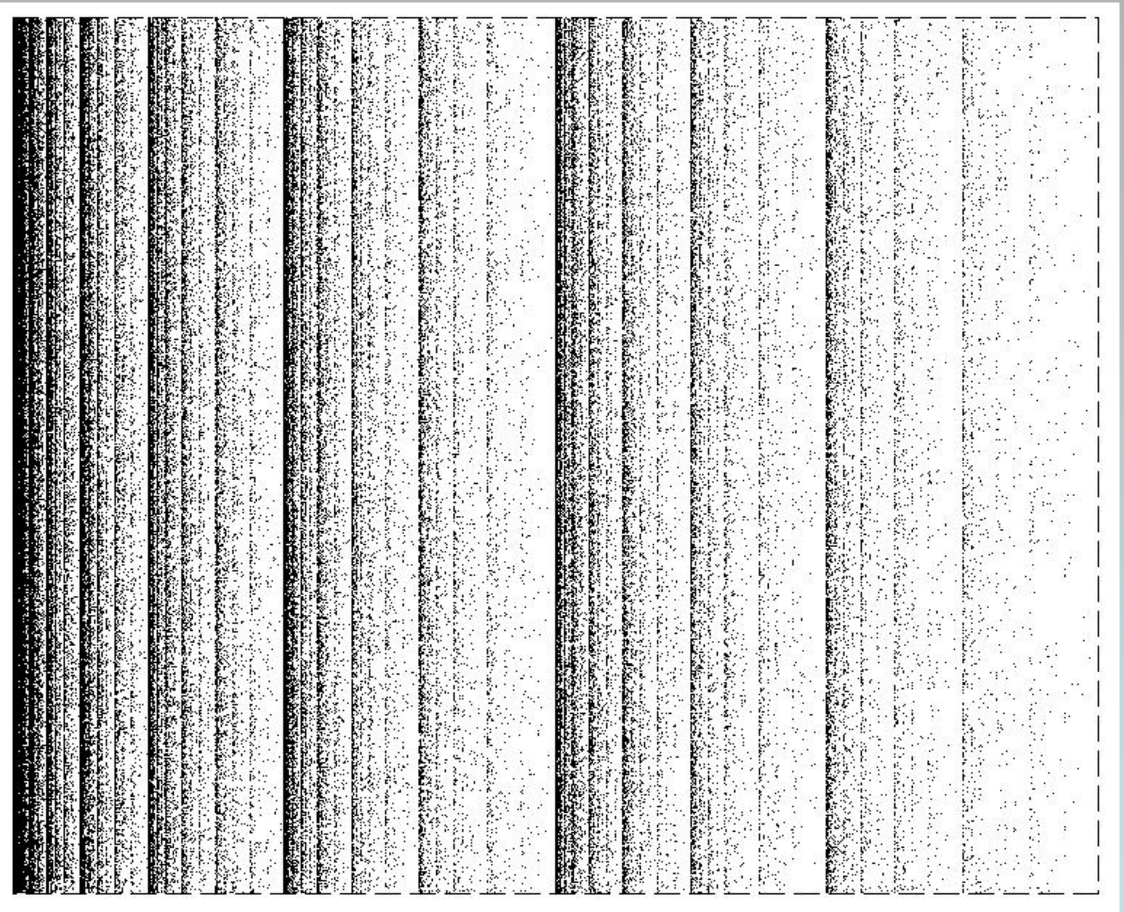




Game sequence: 2 or 3

↓

1 3



4 2

↑

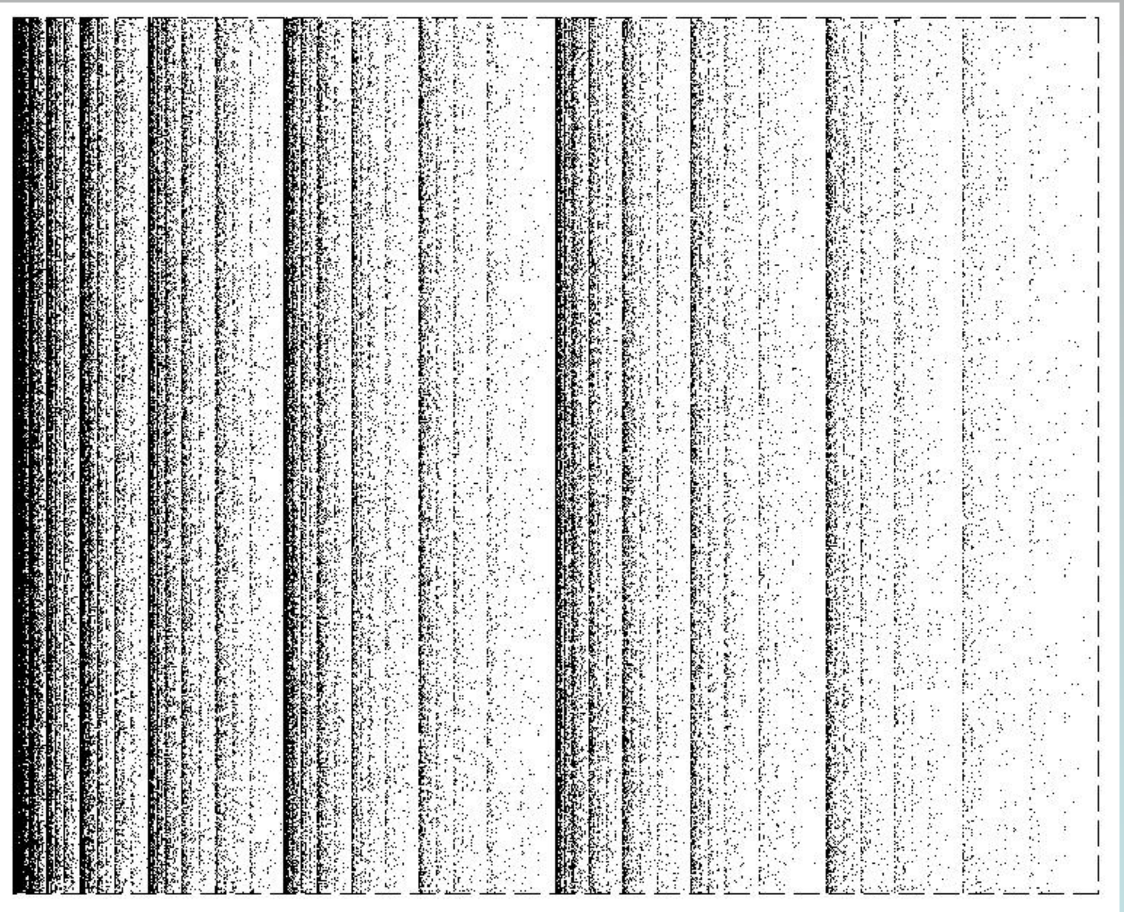
1/2

22 or 33 or 23 or 32

↓

1

3

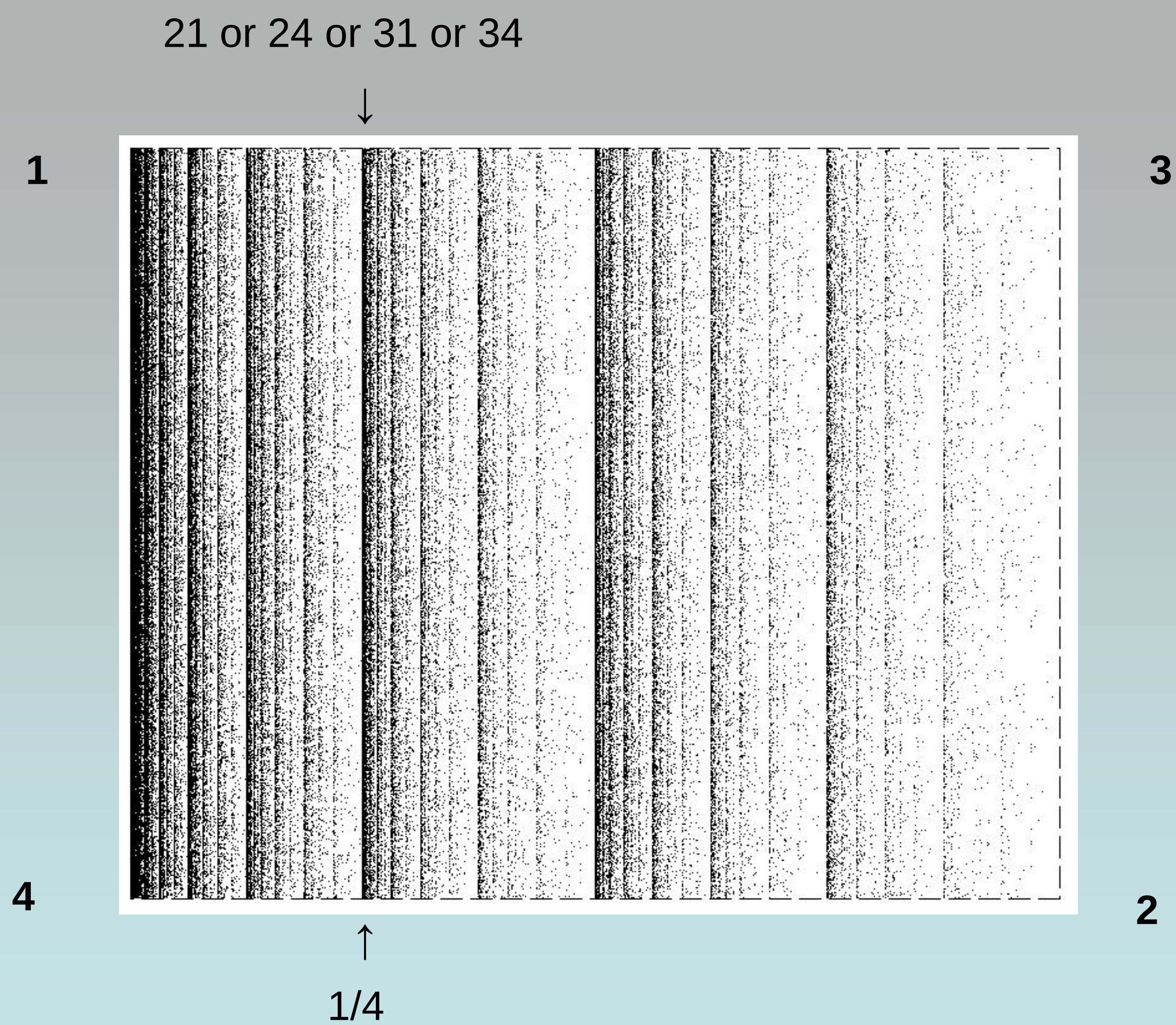


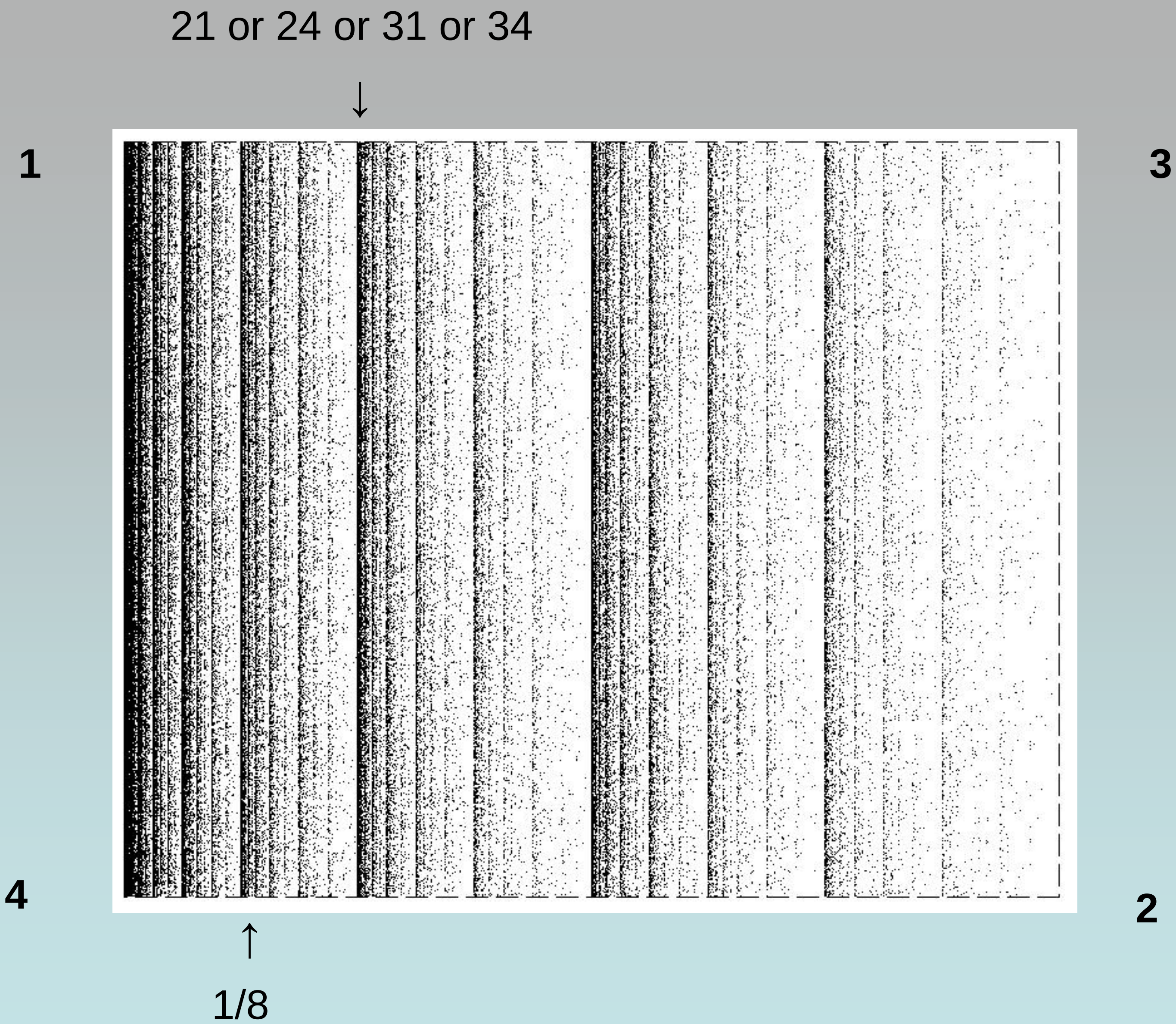
4

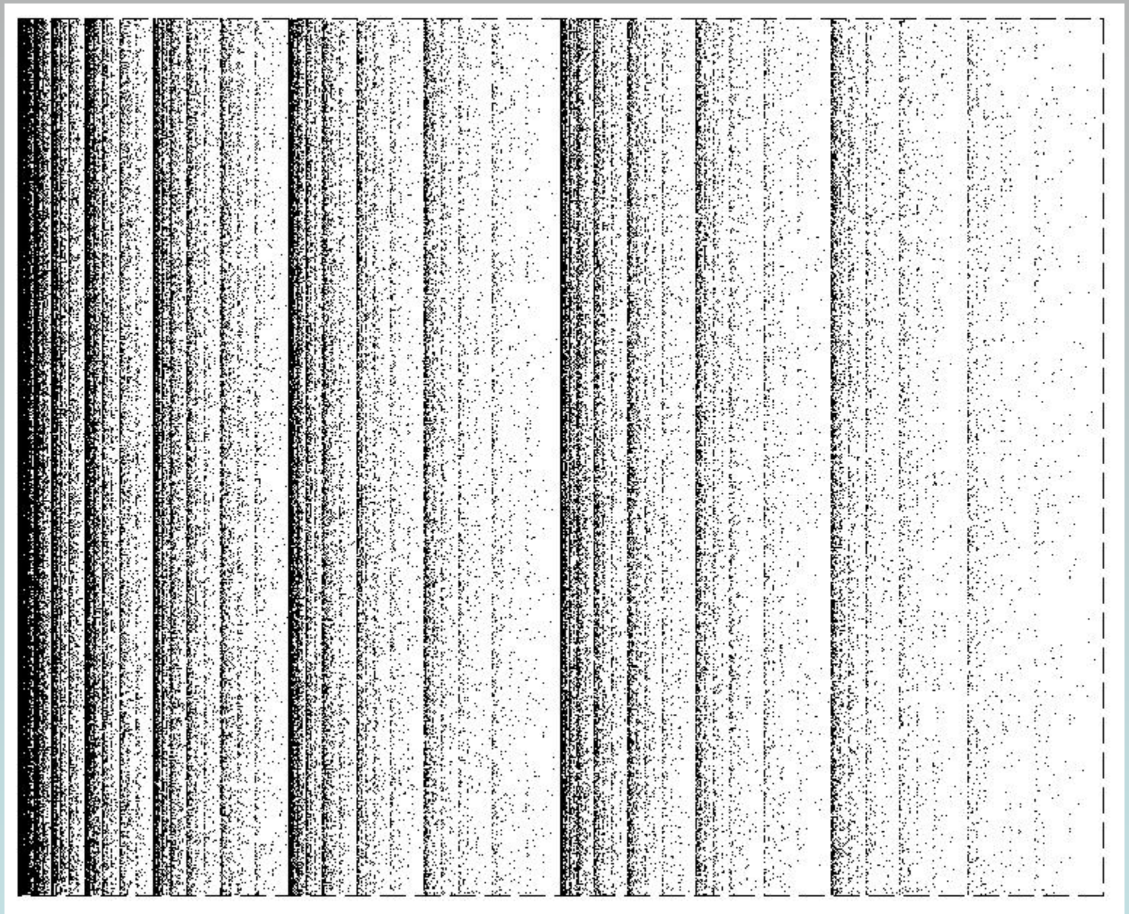
2

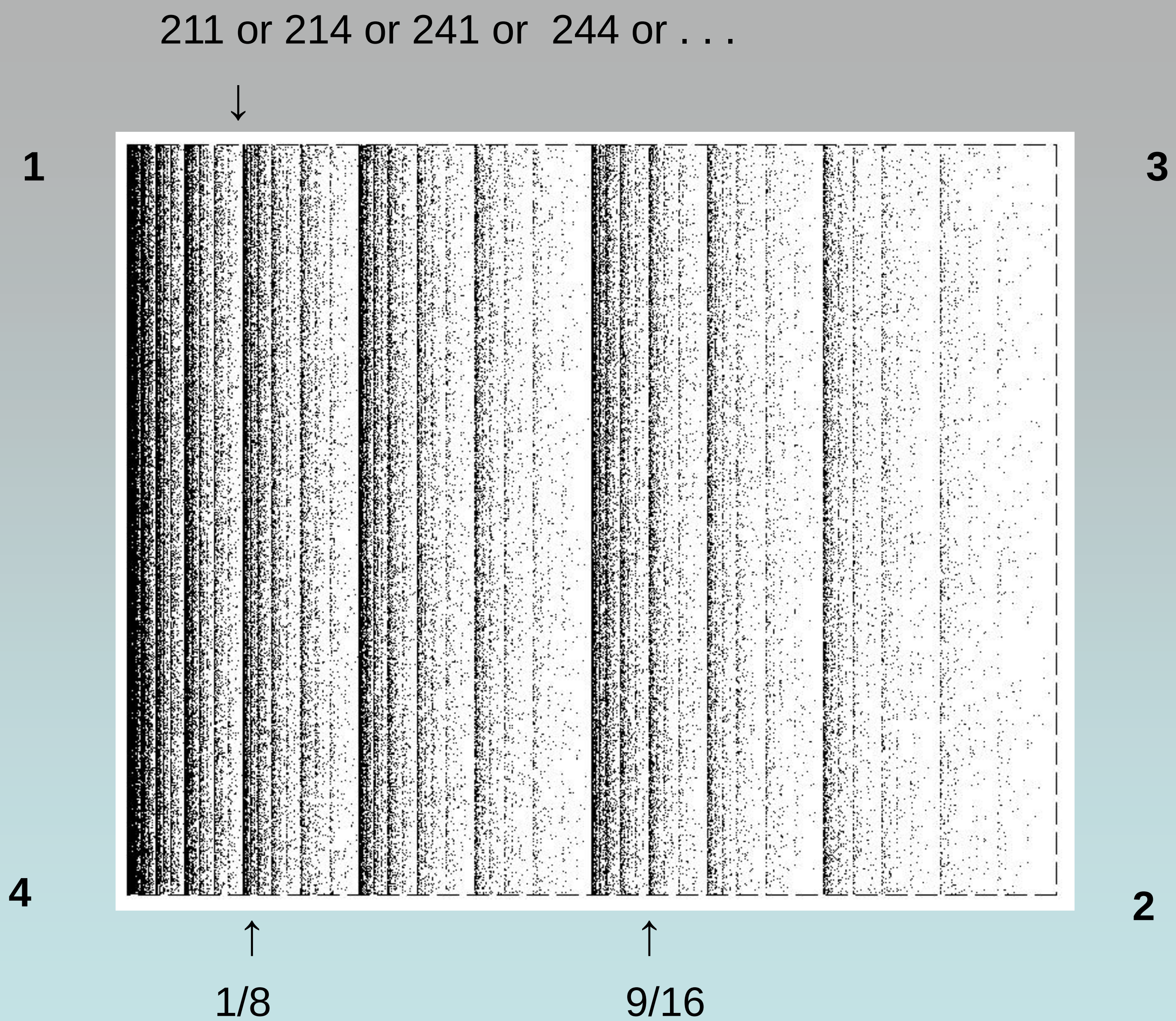
↑

3/4

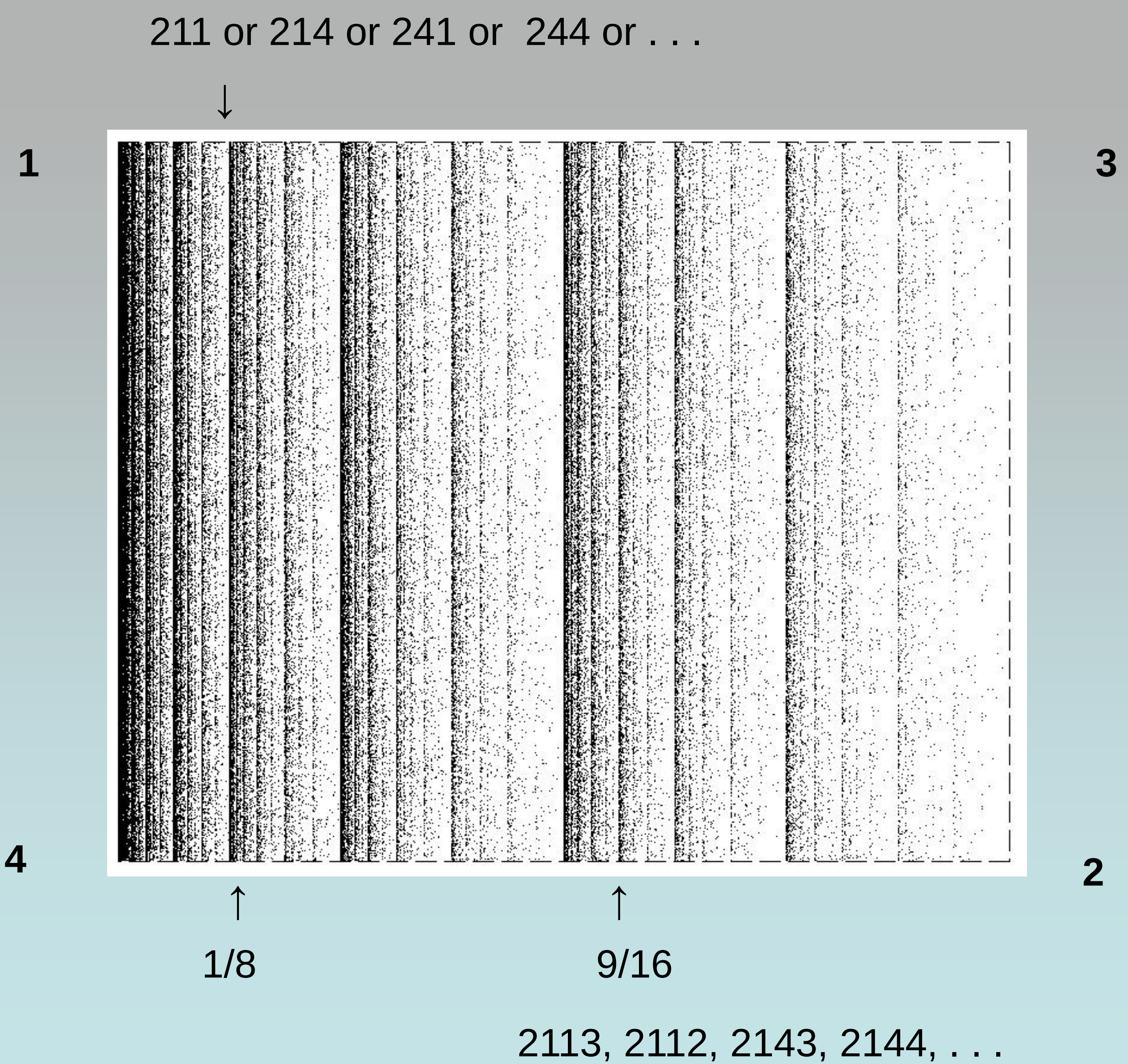

21 or 24 or 31 or 34
↓
1
3
4
2
↑
1/4


21 or 24 or 31 or 34
1
3
4
2
1/8


211 or 214 or 241 or 244 or . . .
↓
1
3
4
2
↑
1/8

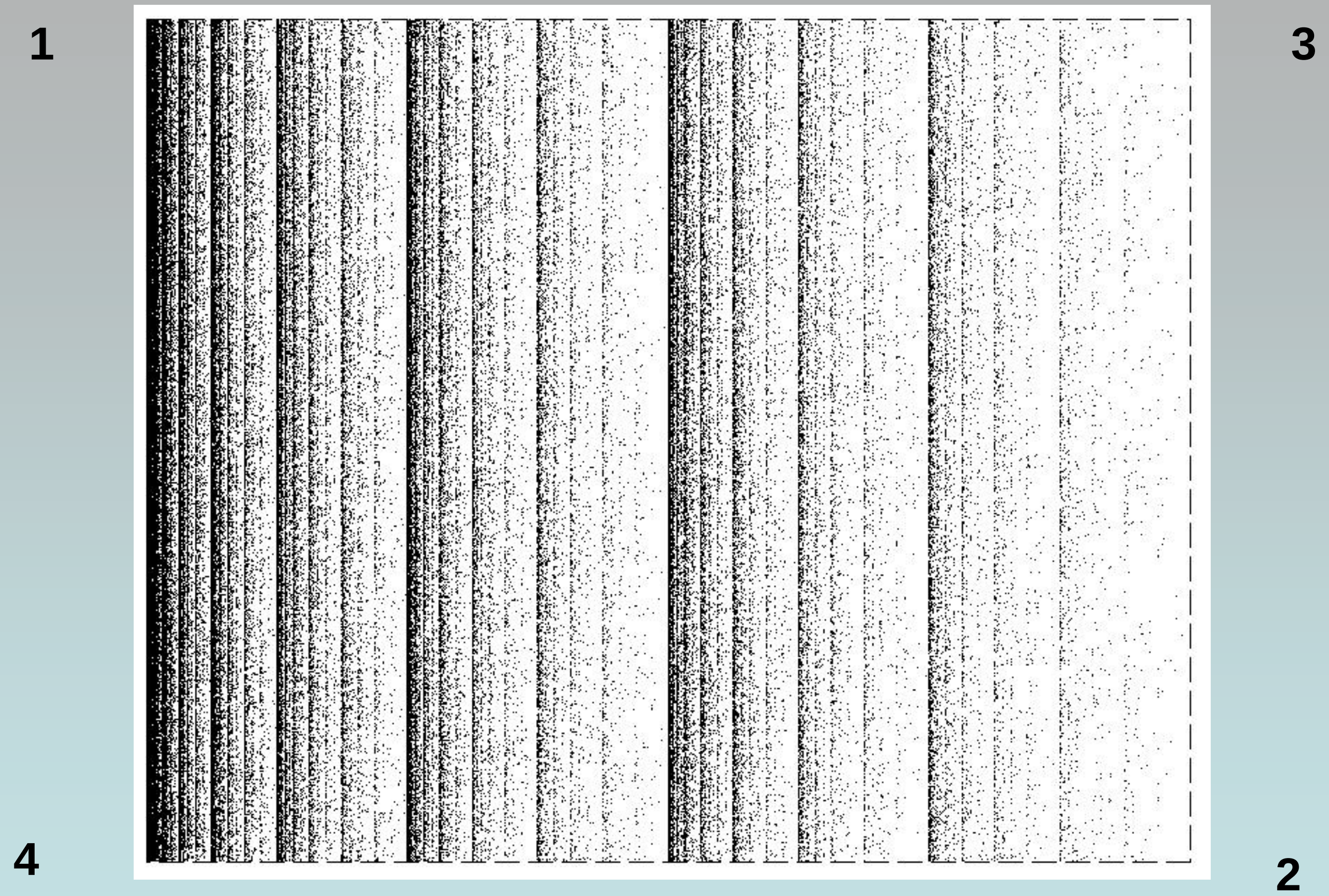


211 or 214 or 241 or 244 or . . .
↓
1
3
4
2
↑
1/8
↑
9/16


211 or 214 or 241 or 244 or . . .
1
3
4
2
1/8
9/16
2113, 2112, 2143, 2144, . . .

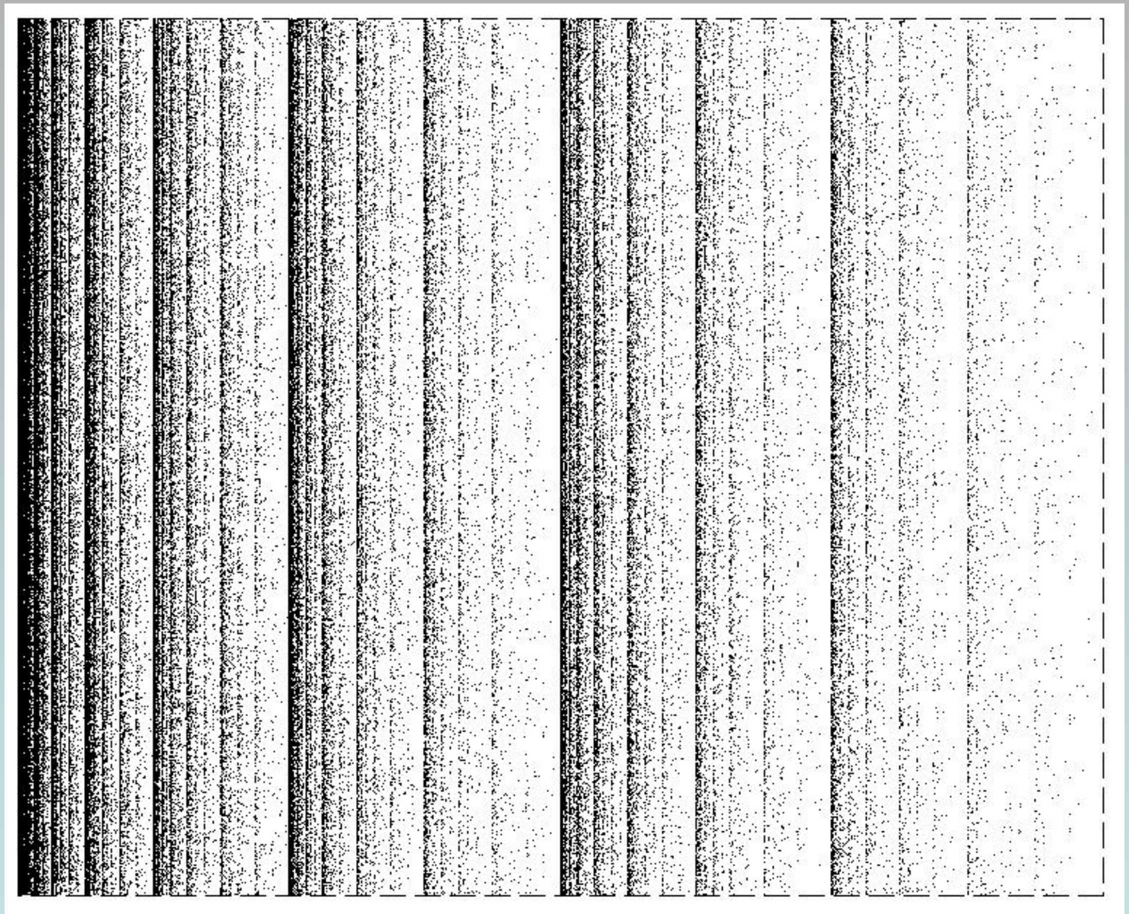
Why so few points in this region?

↓



Why so few points in this region?

↓



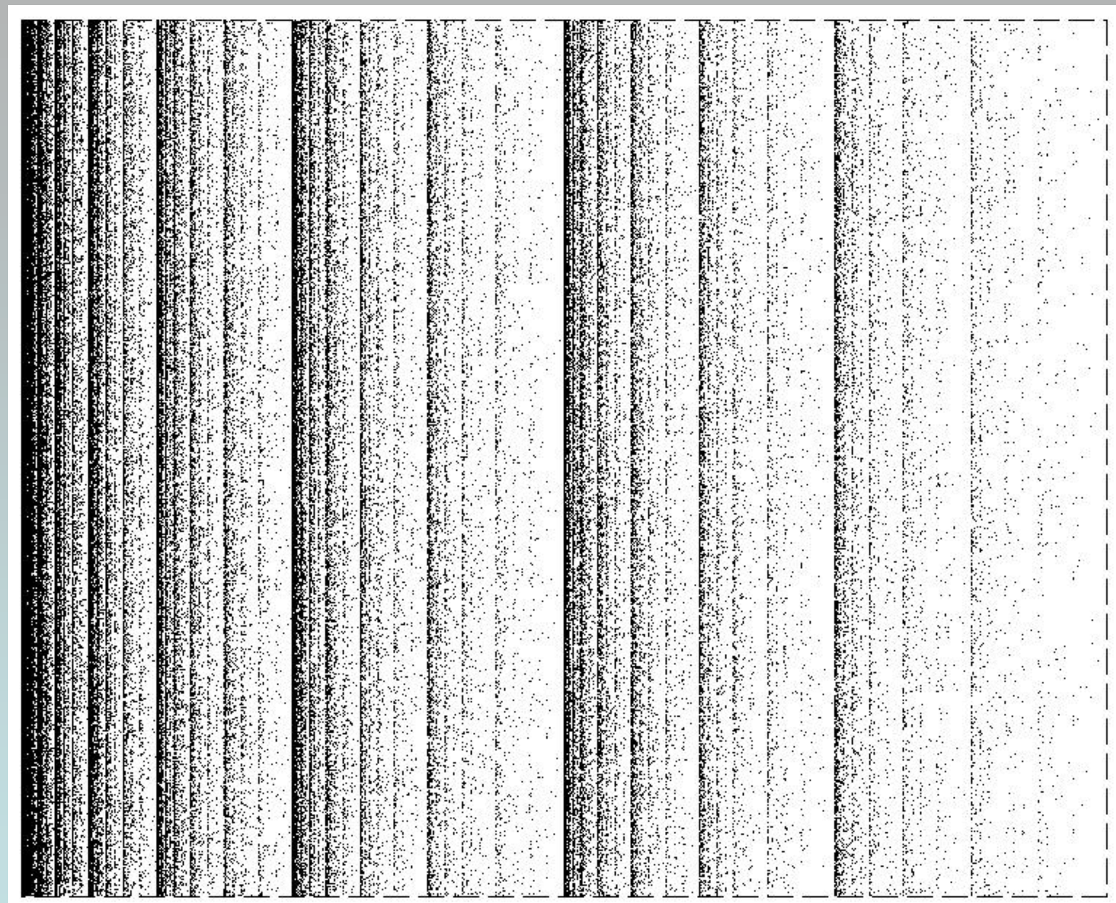
↑

Because points must first go all the way here before going back.

We can estimate the relative intensities of the lines

P=0.4 **1**

3 P=0.1



P=0.4 **4**

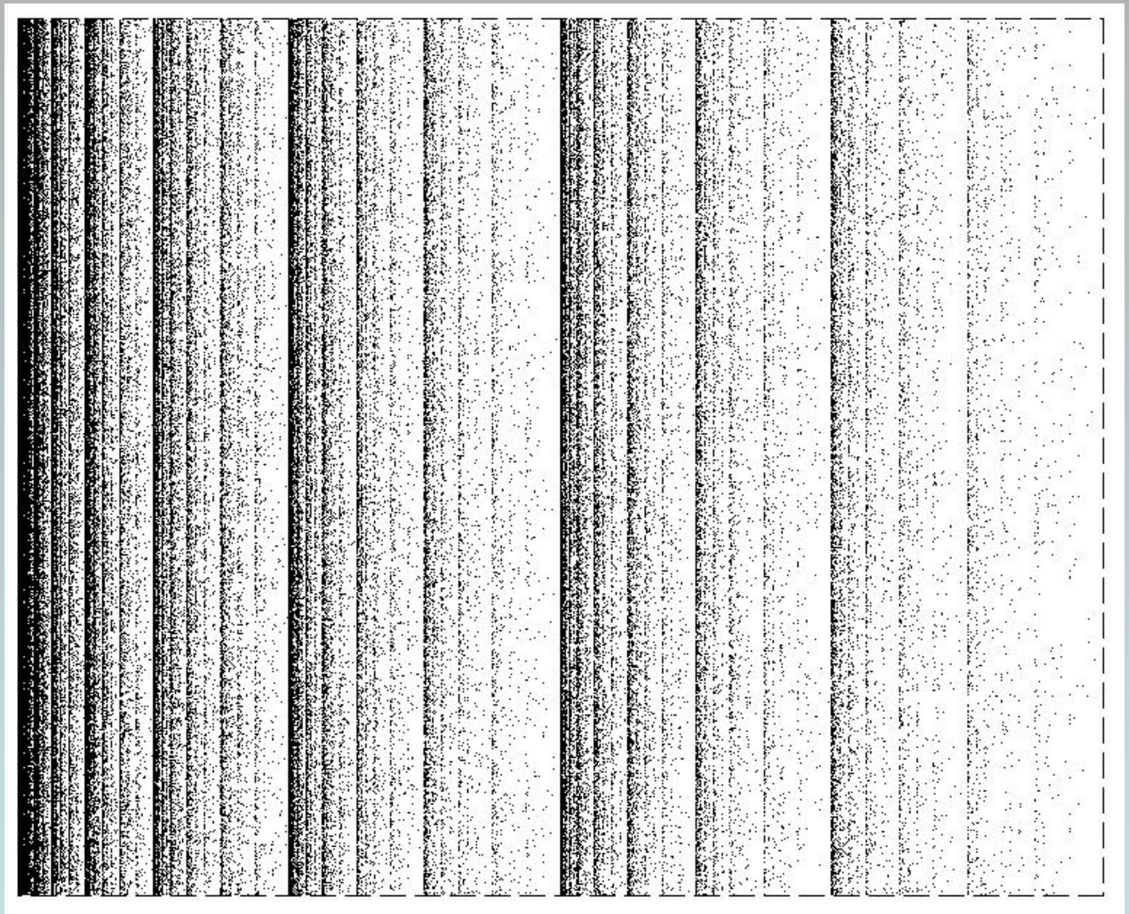
2 P=0.1

Assume a **line** of initial game points here

↓

P=0.4 **1**

3 P=0.1



2 P=0.1

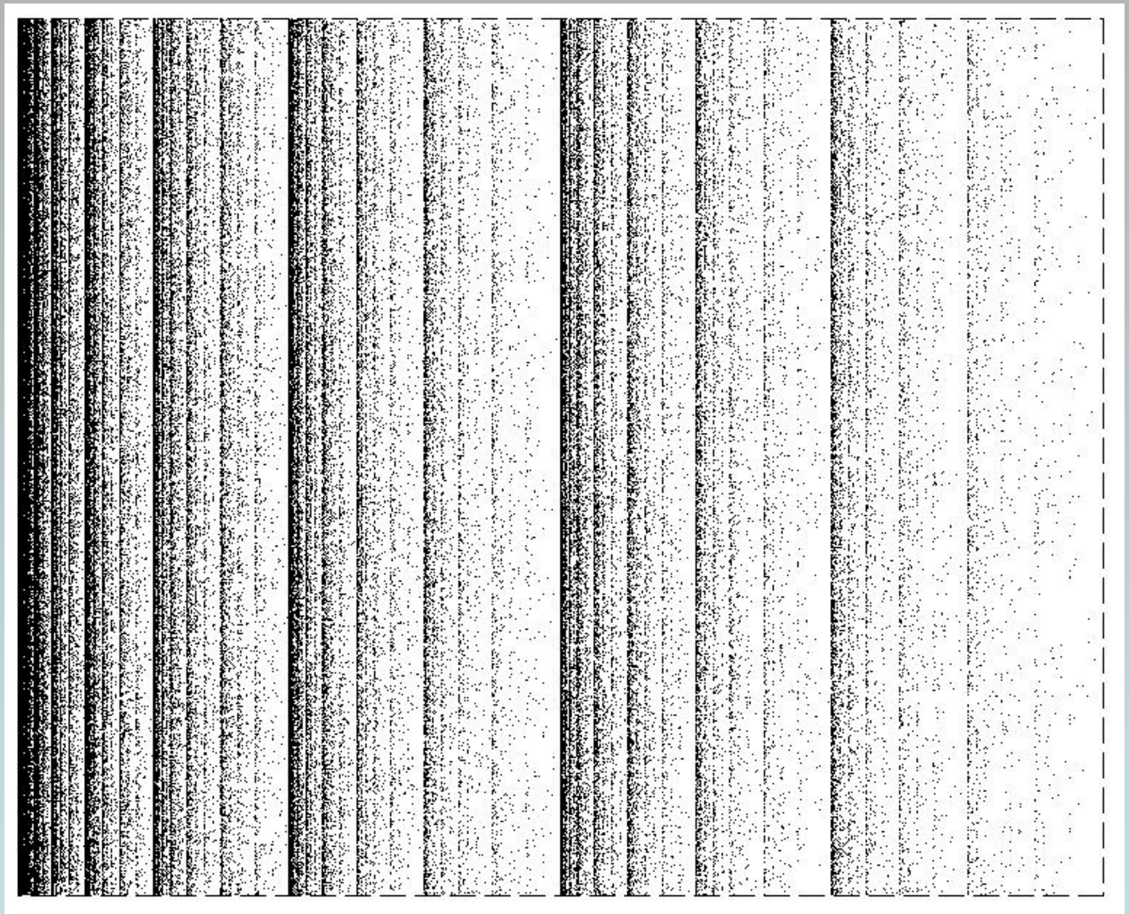
P=0.4 **4**

Assume a **line** of initial game points here

↓

P=0.4 **1**

3 P=0.1



P=0.4 **4**

2 P=0.1

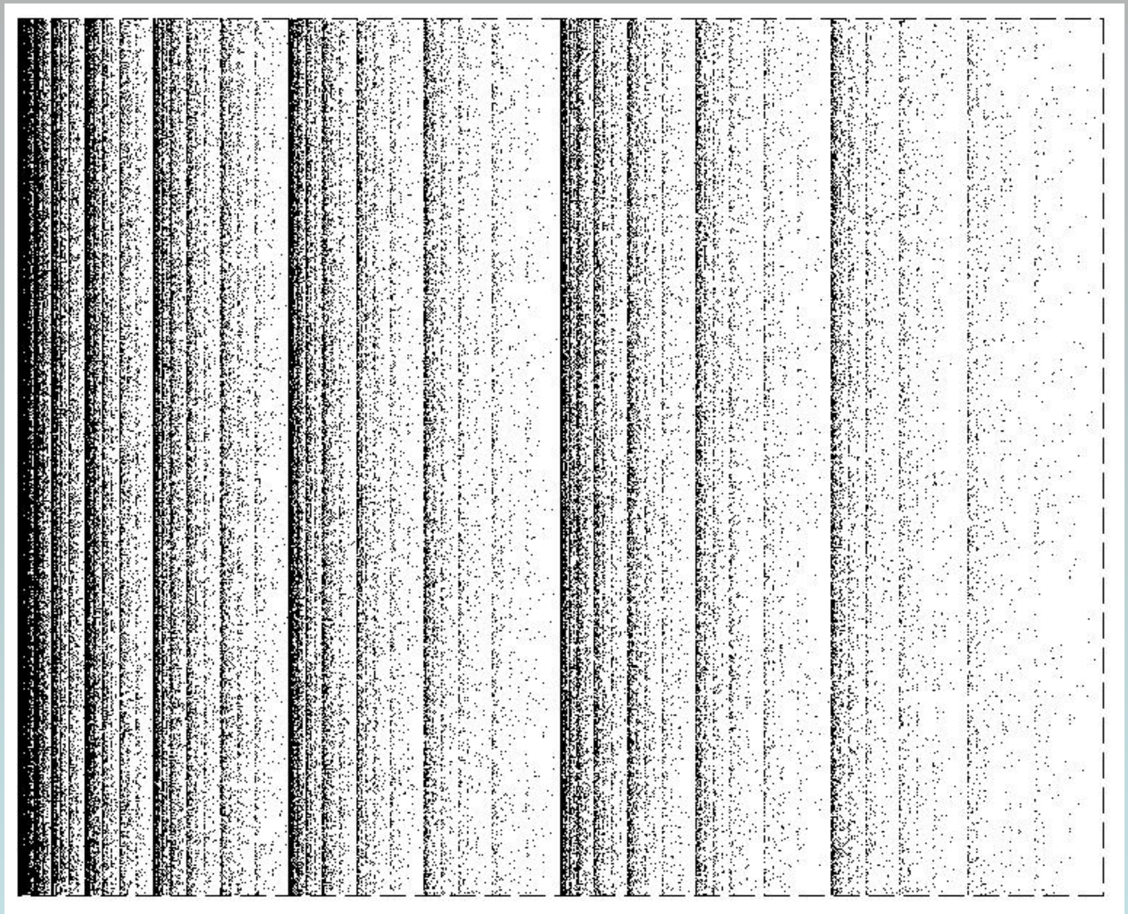
Then we play the chaos game for ***each*** of these points.

Game sequence: 2 or 3

↓

P=0.4 **1**

3 P=0.1



P=0.4 **4**

2 P=0.1

↑

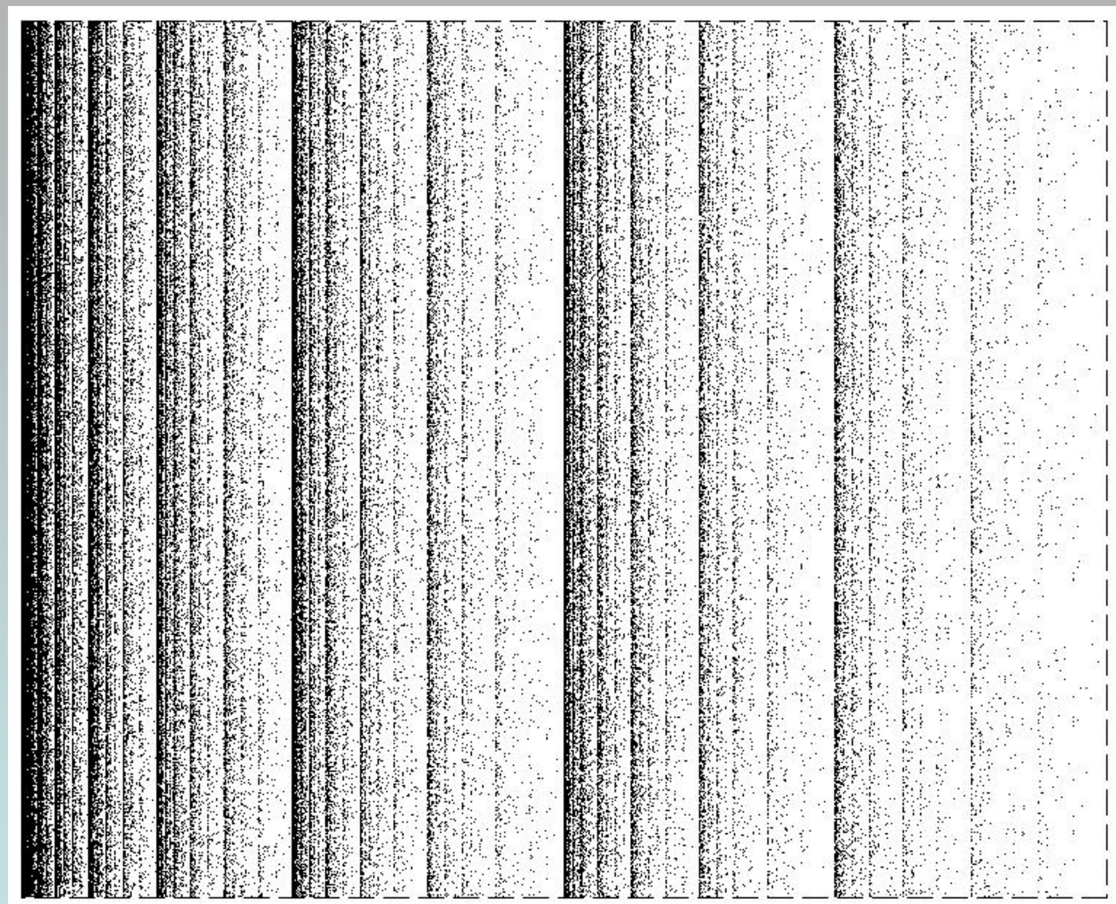
P = 0.1+0.1=0.2

Game sequence: 2 or 3

↓

P=0.4 **1**

3 P=0.1



P=0.4 **4**

2 P=0.1

↑

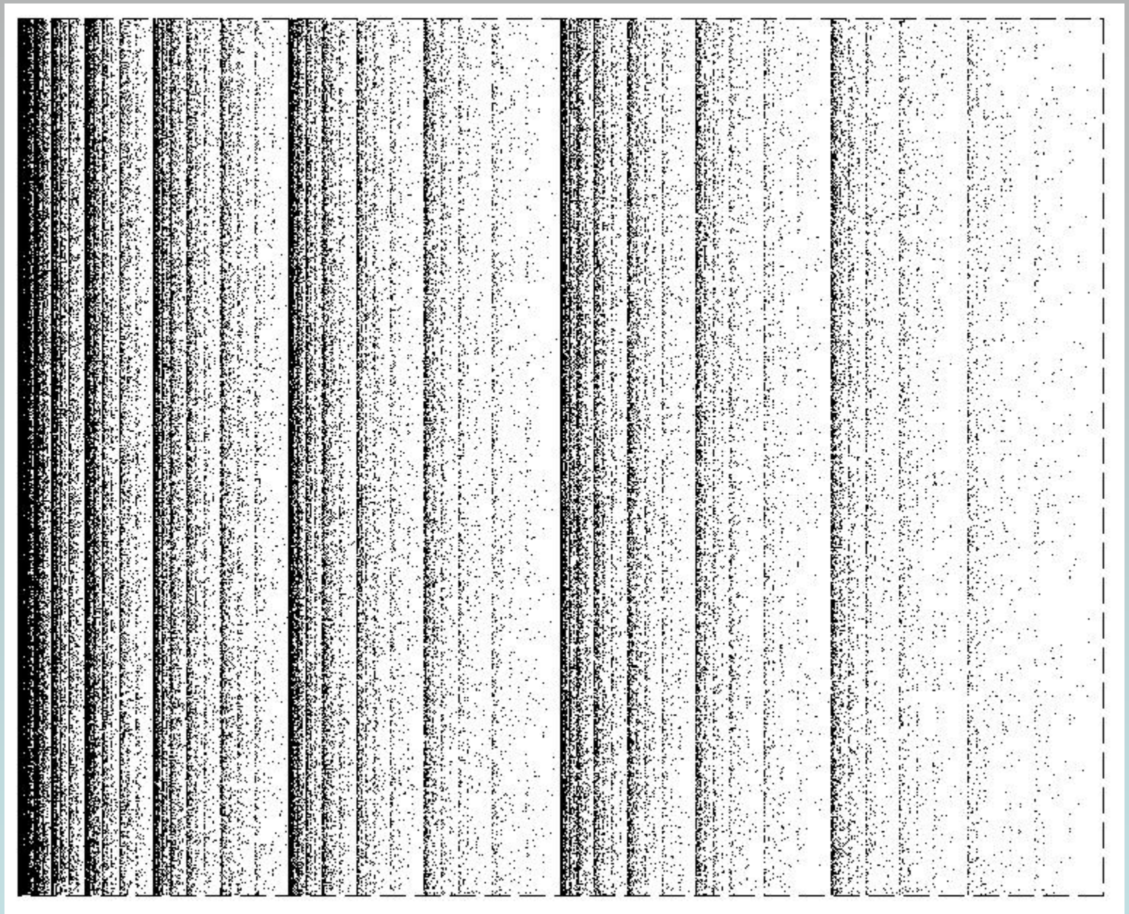
P = 0.1+0.1=0.2
This line is (approximately) 20% as dark as the initial line

22 or 33 or 23 or 32

↓

P=0.4 **1**

3 P=0.1



P=0.4 **4**

2 P=0.1

↑

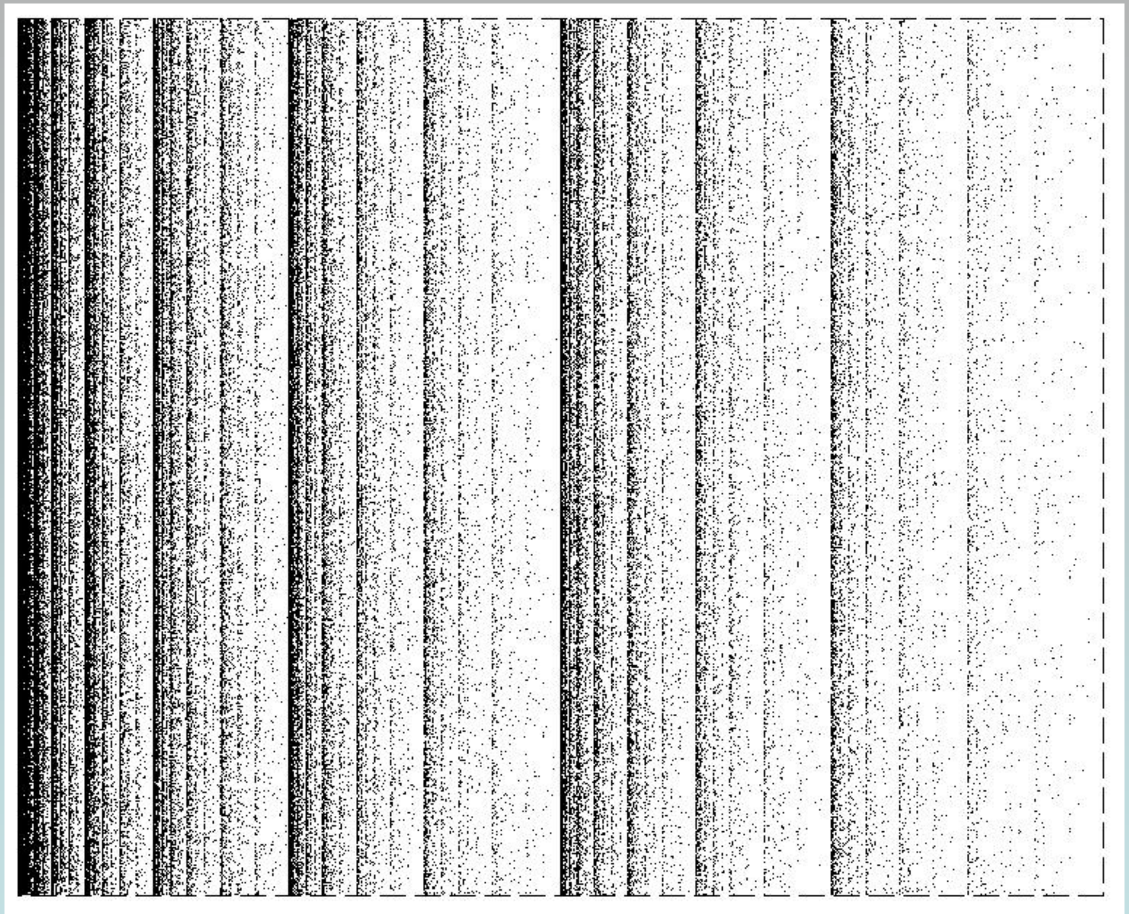
P = 4 (0.1)*(0.1)=0.04

21 or 24 or 31 or 34

↓

P=0.4 **1**

3 P=0.1



P=0.4 **4**

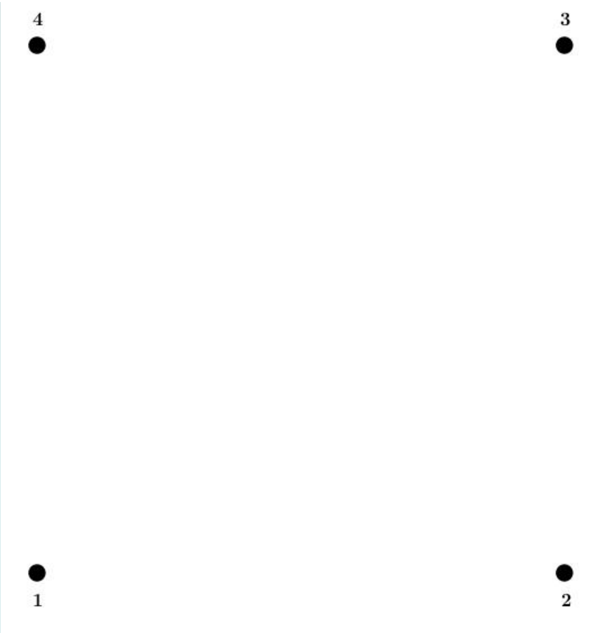
2 P=0.1

↑

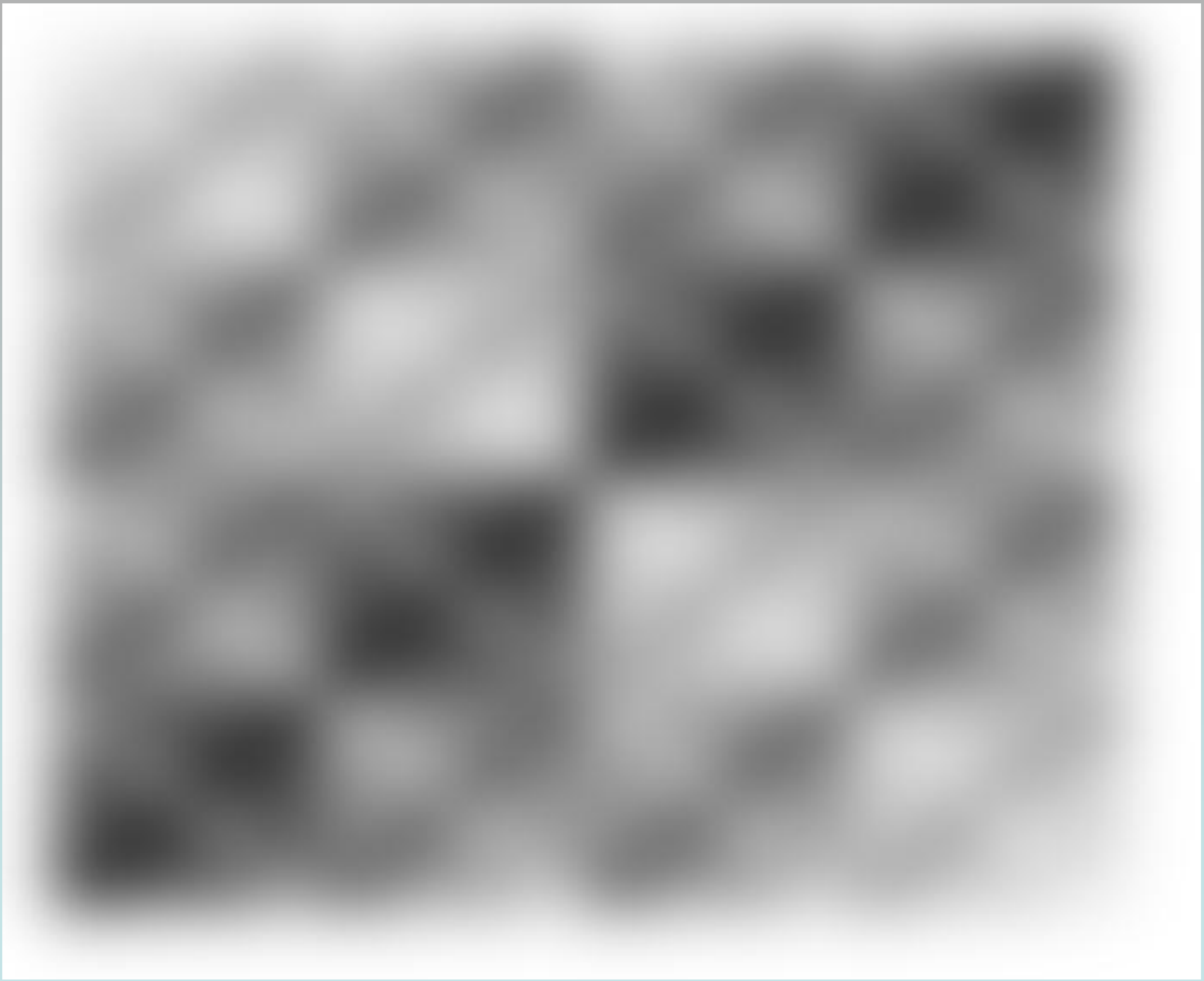
P = 4 (0.1)*(0.4)=0.16

Unequal probabilities (again);

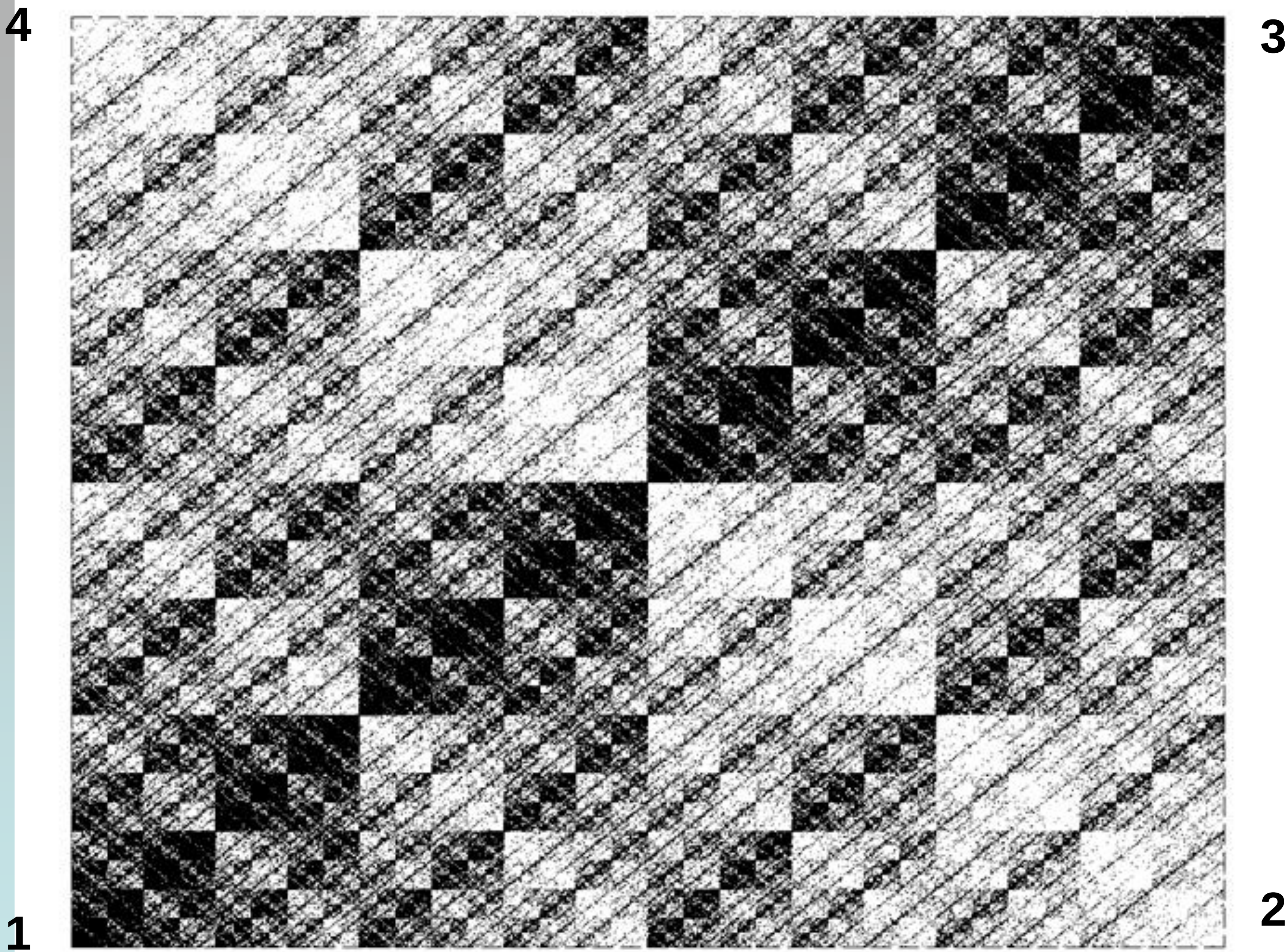
- four pins at the corners of a square
- choose random number $\mathbf{s_i}$ from $\{1, 2, 3, 4\}$ <u>**but this time choose 1 and 3 40% of the time each, and 2 and 4 10% of the time each**</u>
- move $1/2$ distance to pin labelled $\mathbf{s_i}$

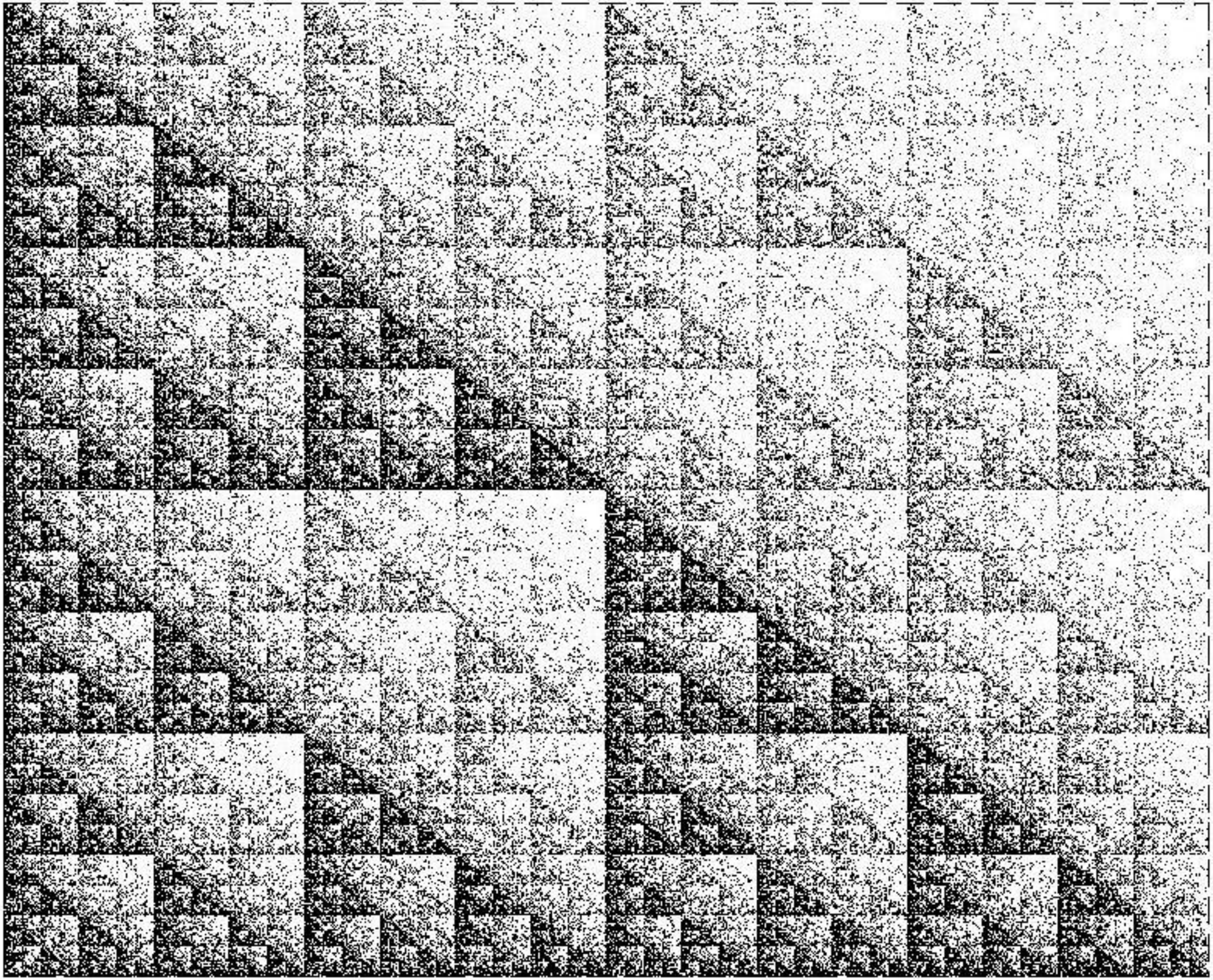


4 3

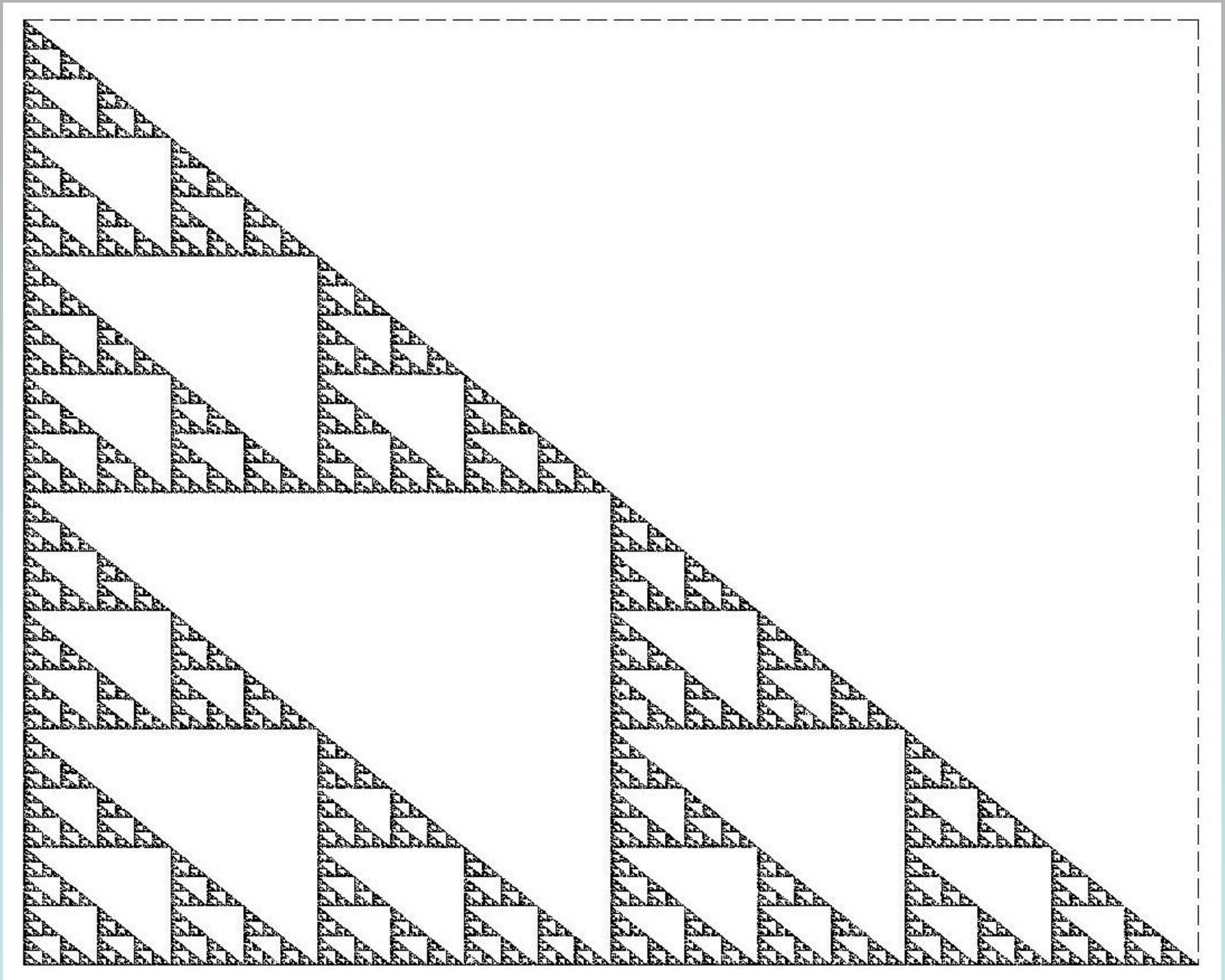


1 2


4
3
1
2



Assign 0 probability to 3



Using other sequences for the chaos game

- random sequences
- non-random, ‘naïve’ sequences
- the ‘best’ sequences

What is the shortest game sequence needed to draw the fractal?

That is, what is the most *efficient* game sequence?

Naïve sequence for Sierpinski

(all addresses of length 3)

11111211131211122123131113213321121221322122222323...

Addresses of game points (* are redundant!);

1 · ·	131	132	233	*231	232	333	*331	*332
11·	213	313	*123	223	323	133	*233	*333
111	*121	*131	*112	*122	*132	*113	*123	*133
*111	*112	113	*211	*212	*213	*311	*312	*313
*111	*211	*311	*121	*221	*321	*131	*231	*331
211	221	231	*212	222	*232	*213	*223	*233
121	122	123	*221	*222	*223	*321	*322	*323
112	212	312	*122	*222	*322	*132	*232	*332
311	321	331	*312	322	332	*313	*323	*333

1 · ·	131	132	233	∗231	232	333	∗331	∗332
11·	213	313	∗123	223	323	133	∗233	∗333
111	∗121	∗131	∗112	∗122	∗132	∗113	∗123	∗133
∗111	∗112	113	∗211	∗212	∗213	∗311	∗312	∗313
∗111	∗211	∗311	∗121	∗221	∗321	∗131	∗231	∗331
211	221	231	∗212	222	∗232	∗213	∗223	∗233
121	122	123	∗221	∗222	∗223	∗321	∗322	∗323
112	212	312	∗122	∗222	∗322	∗132	∗232	∗332
311	321	331	∗312	322	332	∗313	∗323	∗333

Left with the 27 distinct addresses of length 3.

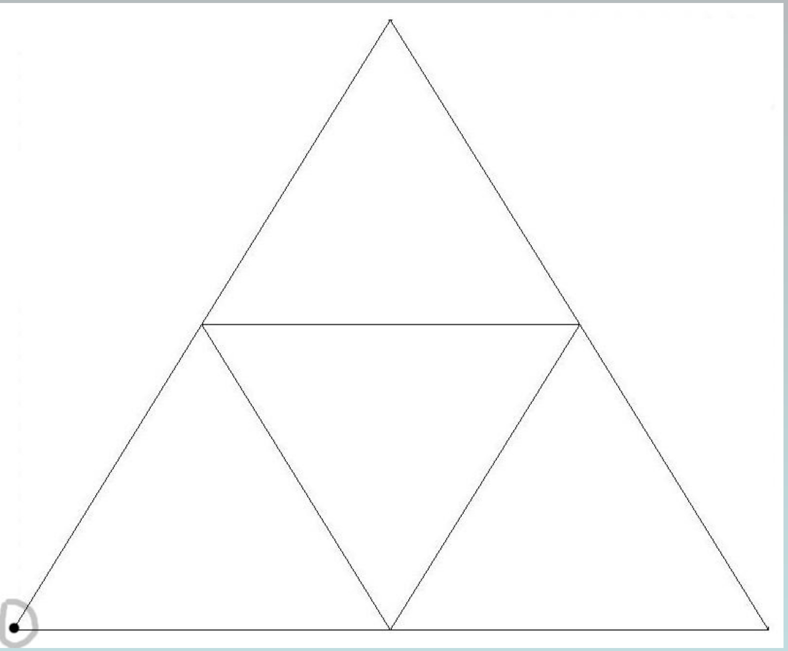
1 · ·	131	132	233	*231	232	333	*331	*332
11·	213	313	*123	223	323	133	*233	*333
111	*121	*131	*112	*122	*132	*113	*123	*133
*111	*112	113	*211	*212	*213	*311	*312	*313
*111	*211	*311	*121	*221	*321	*131	*231	*331
211	221	231	*212	222	*232	*213	*223	*233
121	122	123	*221	*222	*223	*321	*322	*323
112	212	312	*122	*222	*322	*132	*232	*332
311	321	331	*312	322	332	*313	*323	*333

Left with the 27 distinct addresses of length 3.

The ‘best’ sequence;
11121131221231312133222323331

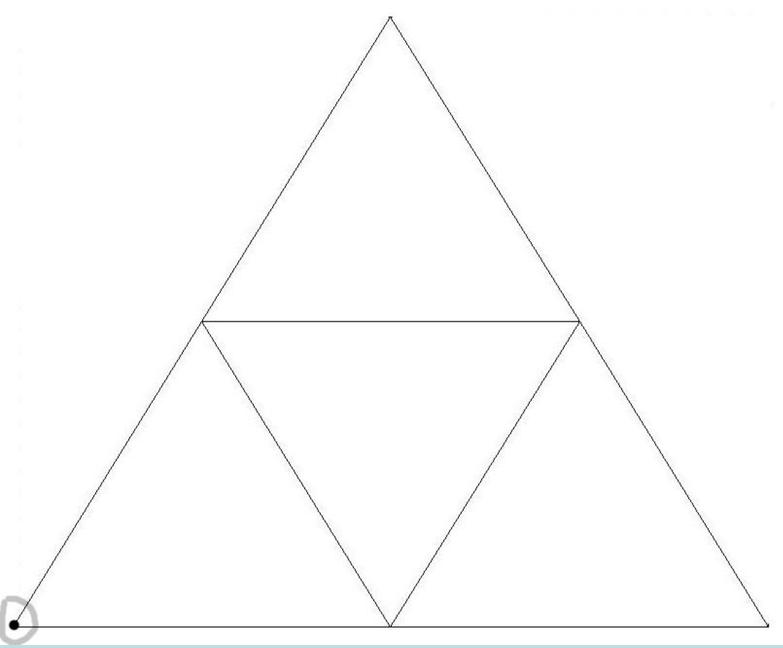
Playing the chaos game with non-random sequences

Playing the chaos game with non-random sequences



- Sierpinski game. Initial game point at bottom left corner.

Playing the chaos game with non-random sequences

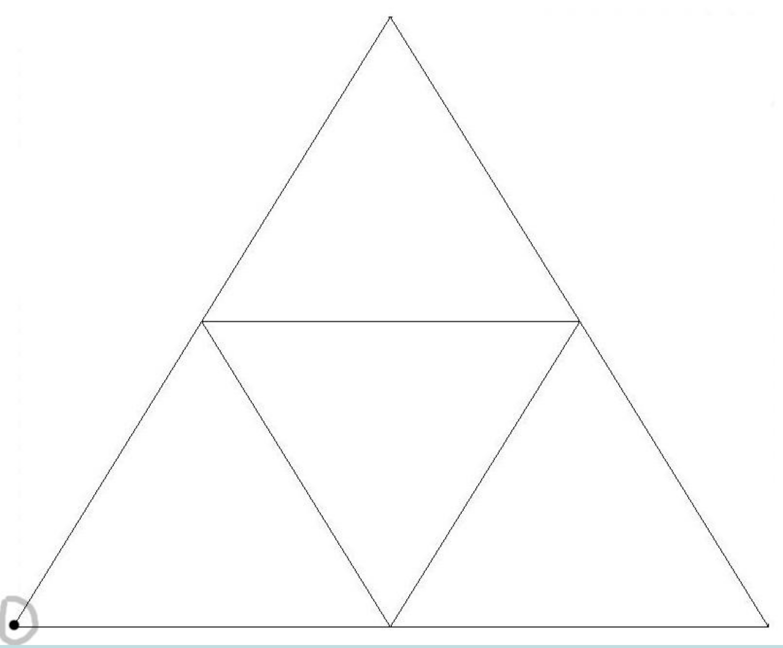


Sierpinski game. Initial game point at bottom left corner.

Game numbers; 123123123123123....

(i.e., 123 repeating).

Playing the chaos game with non-random sequences

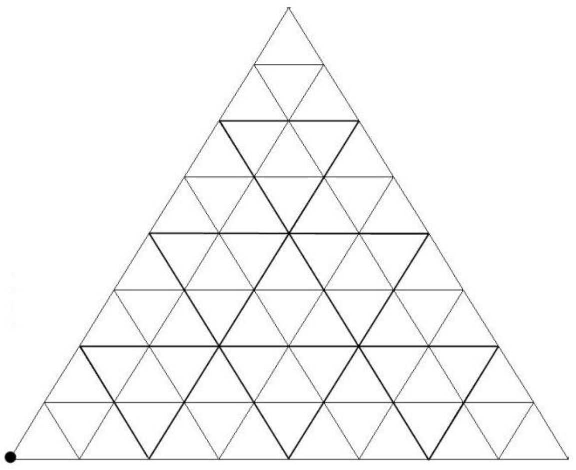


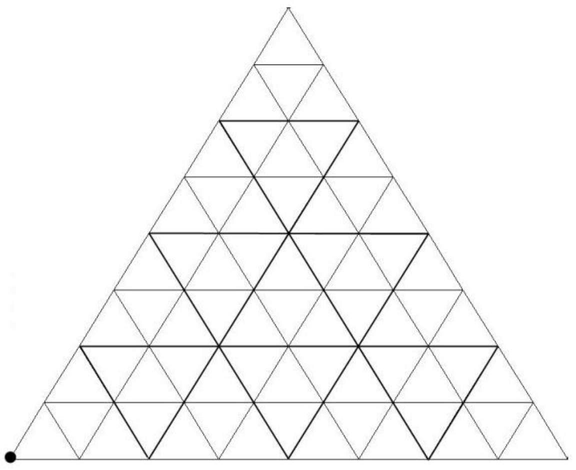
Sierpinski game. Initial game point at bottom left corner.

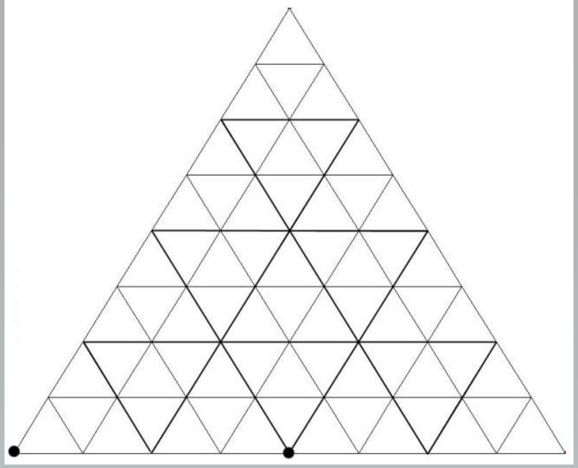
Game numbers; 123123123123123....

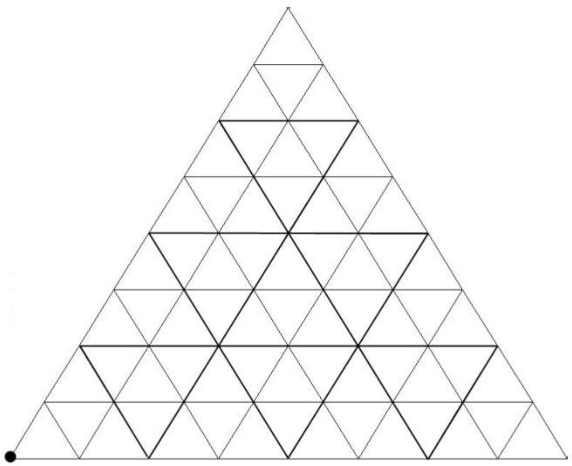
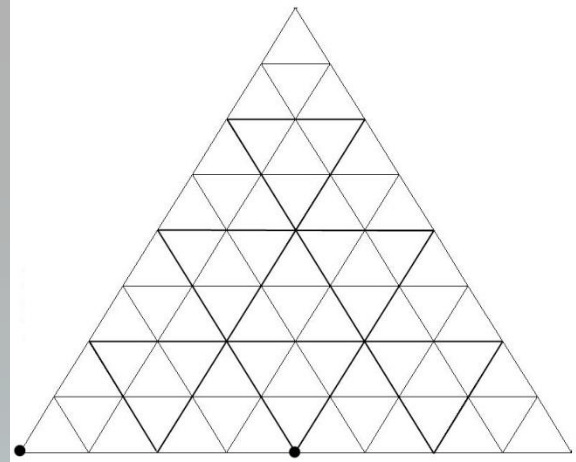
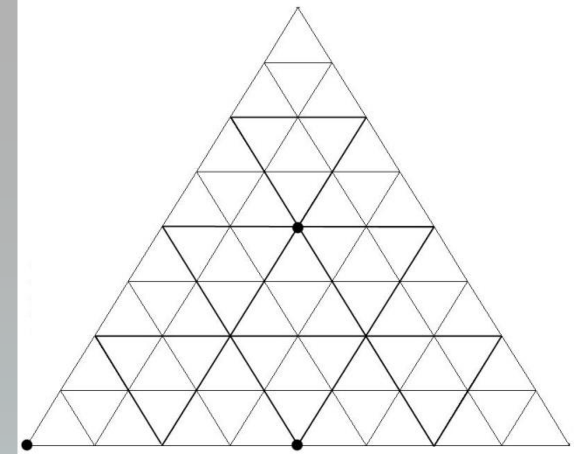
(i.e., 123 repeating).

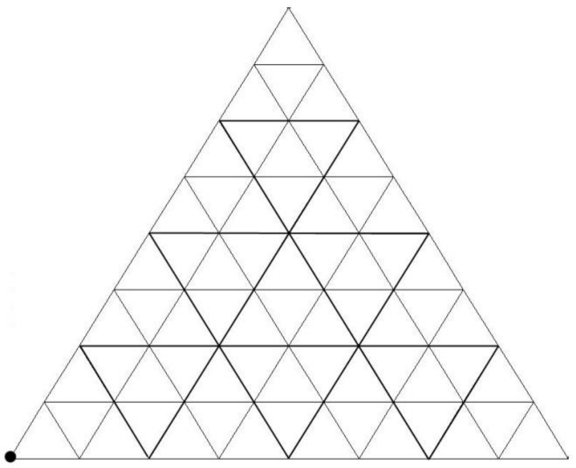
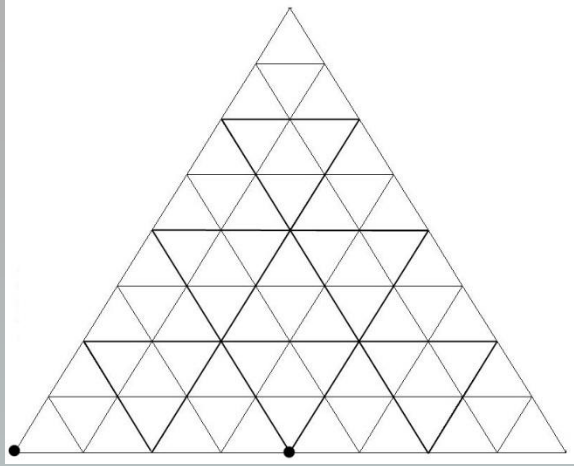
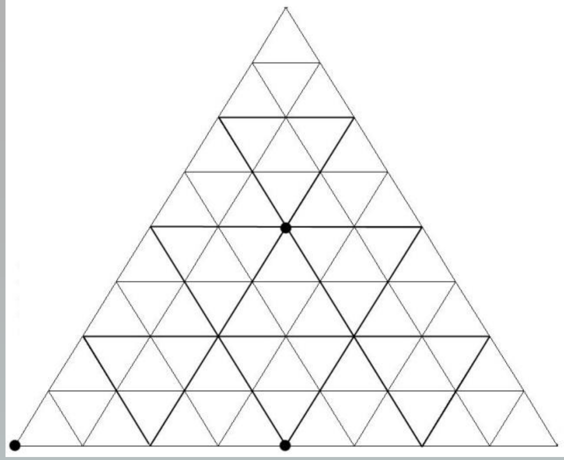
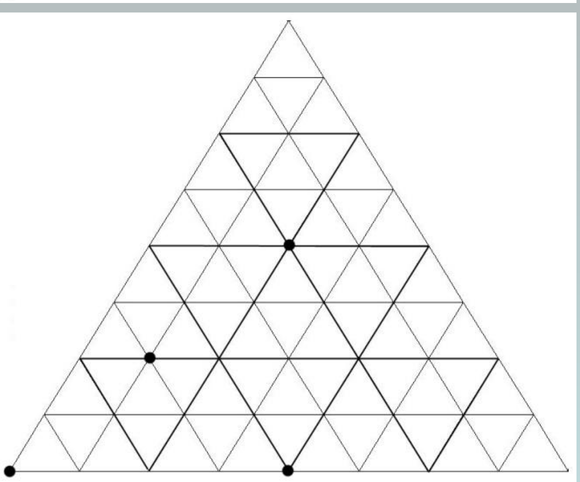
Outcome?

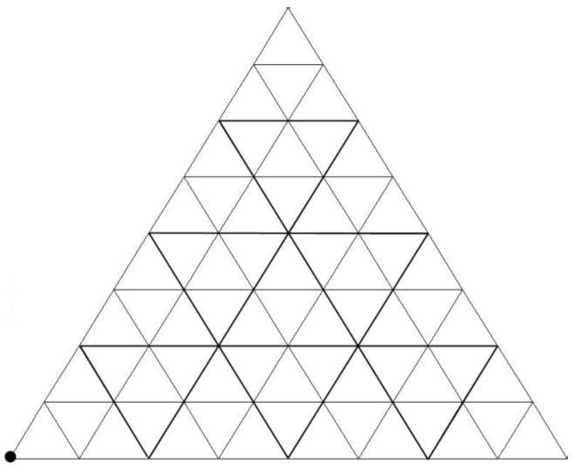


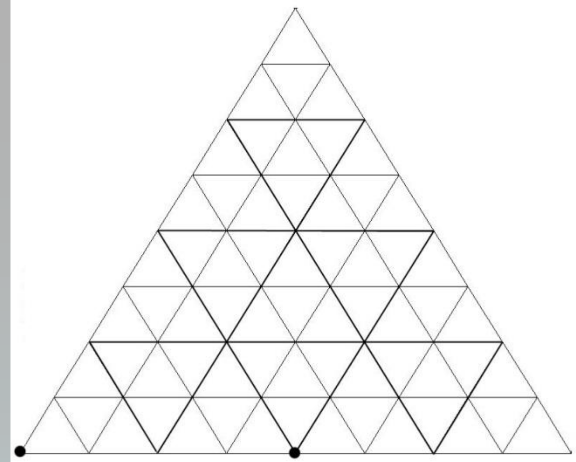


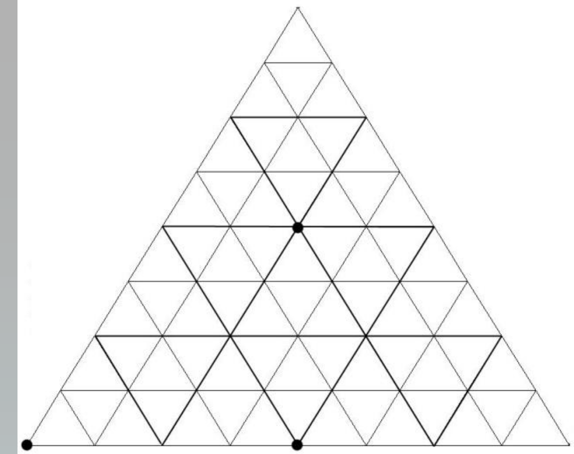


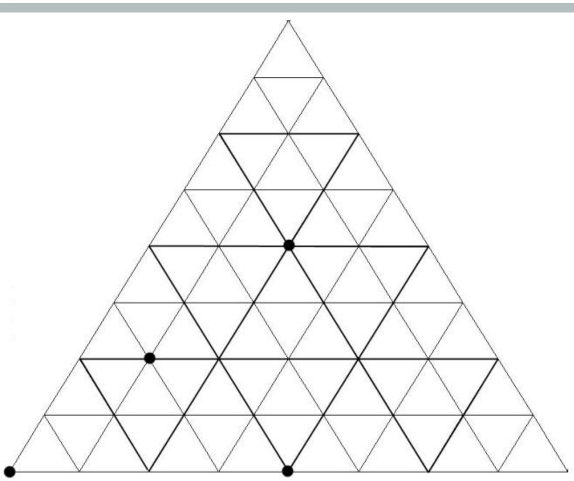




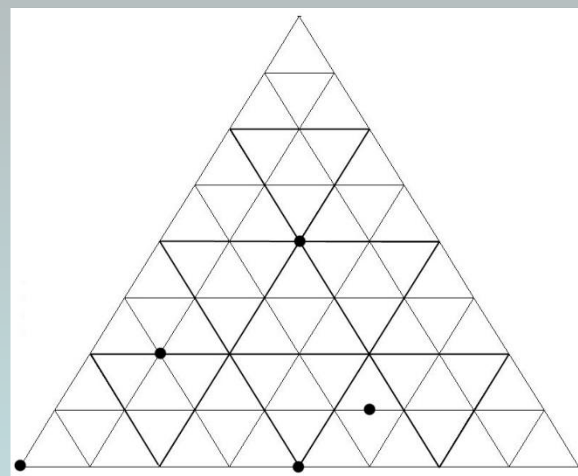





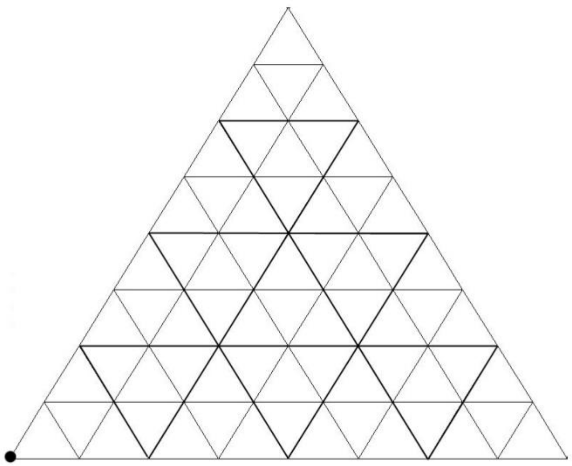
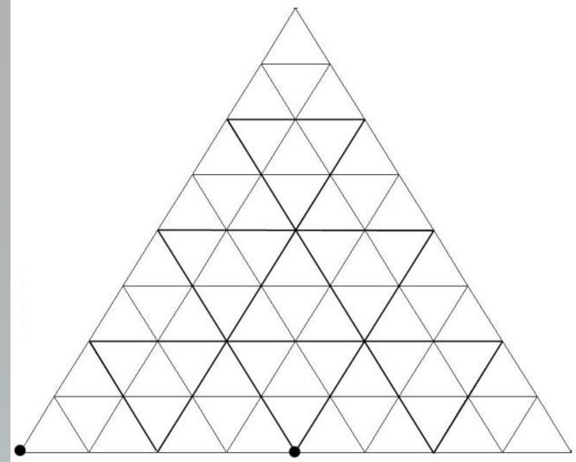
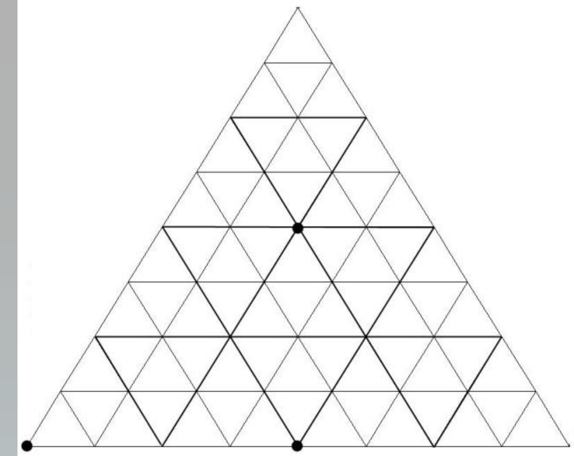
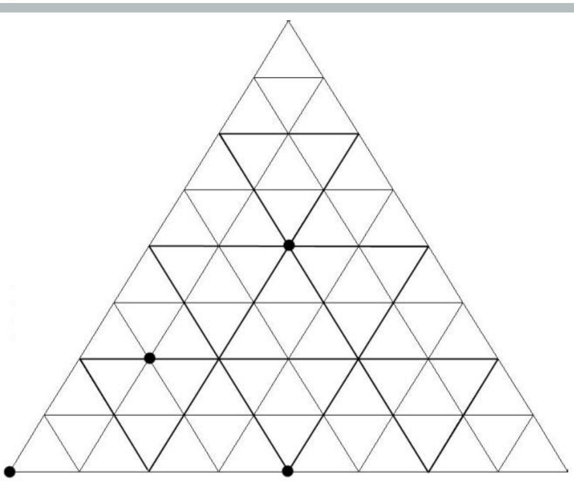
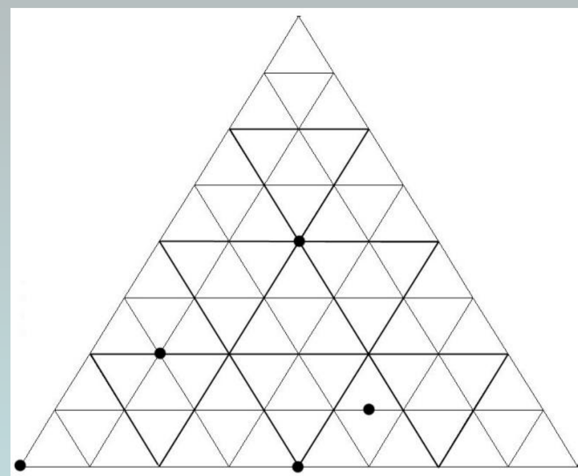
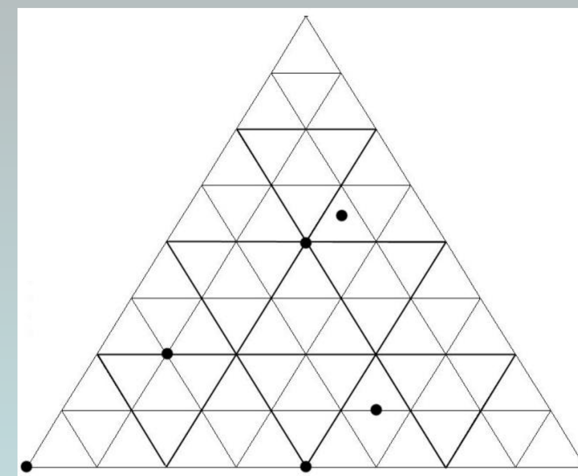


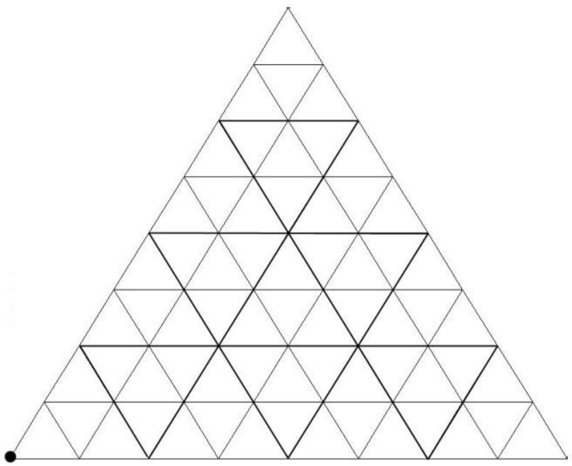


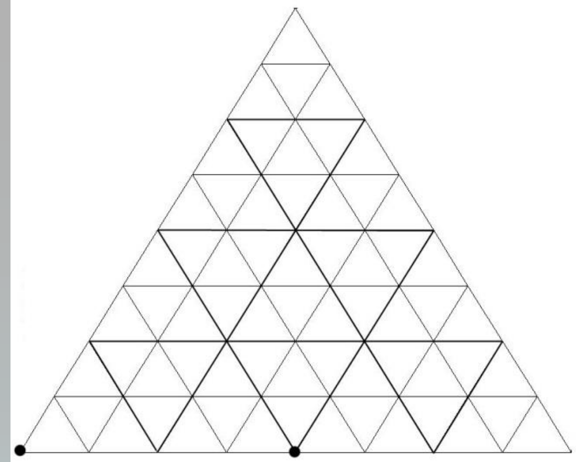


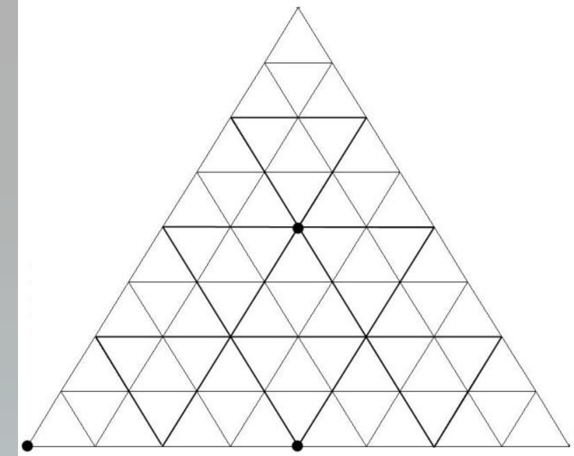


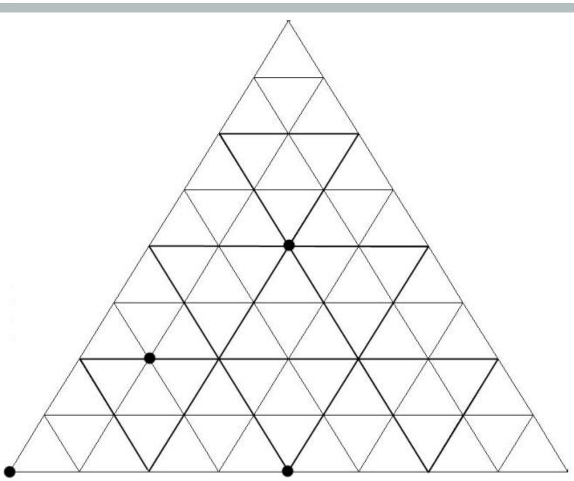


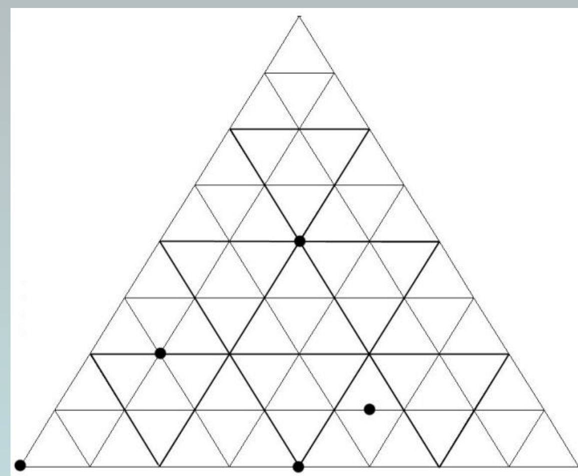







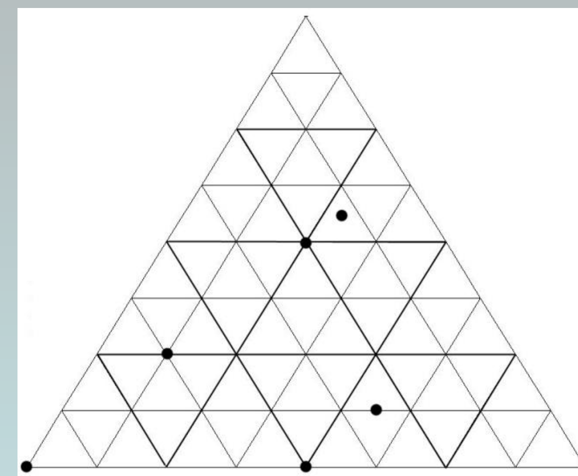


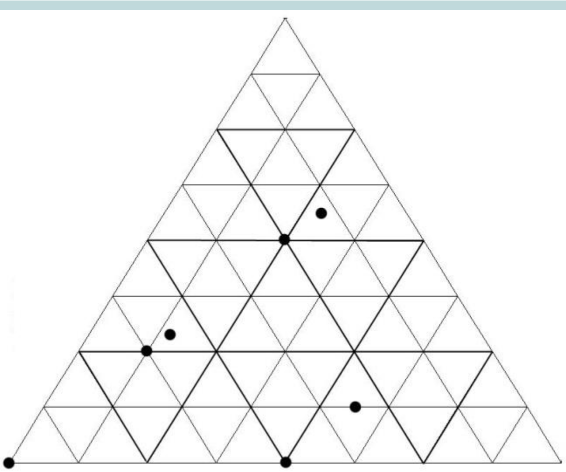


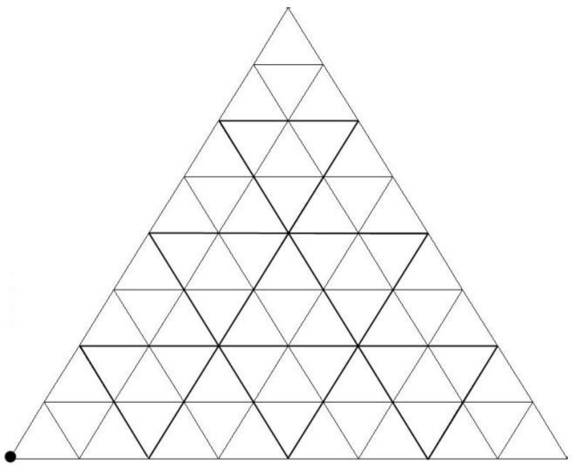
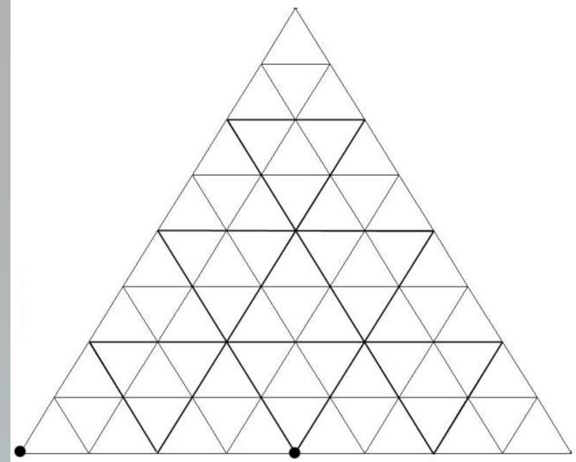
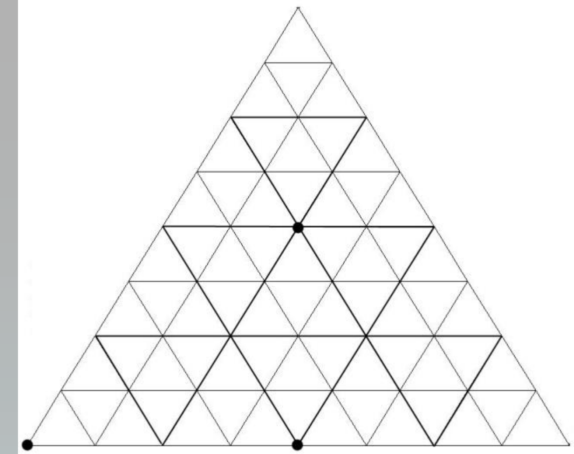
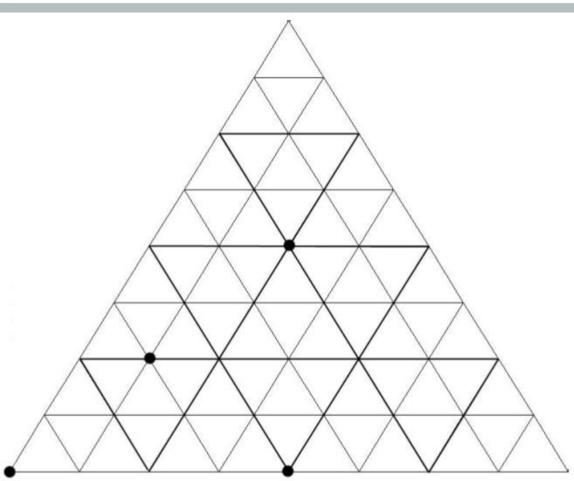
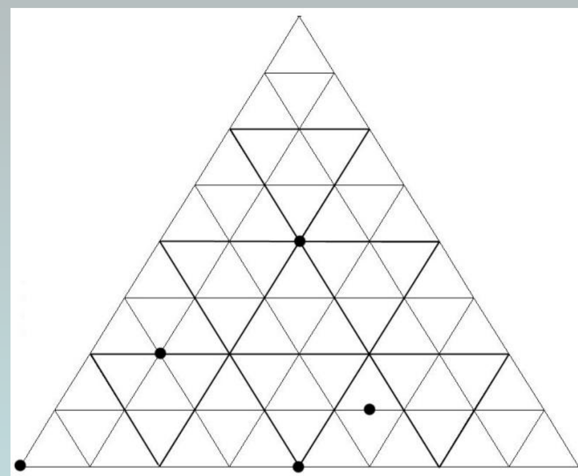
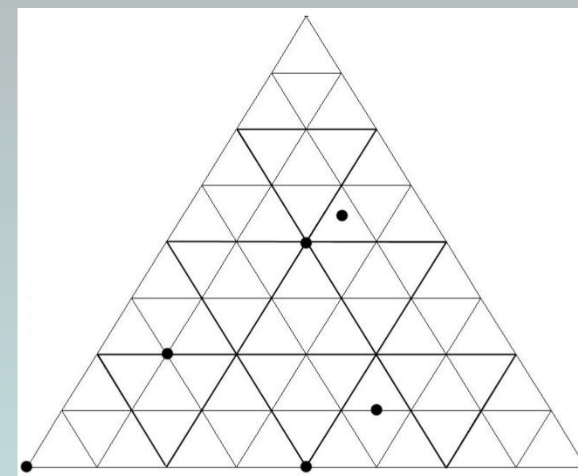
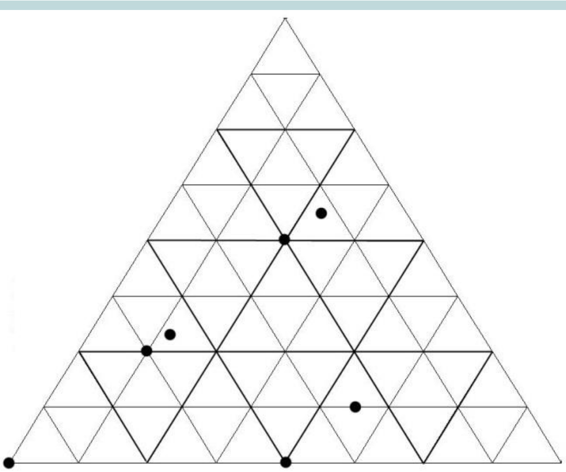
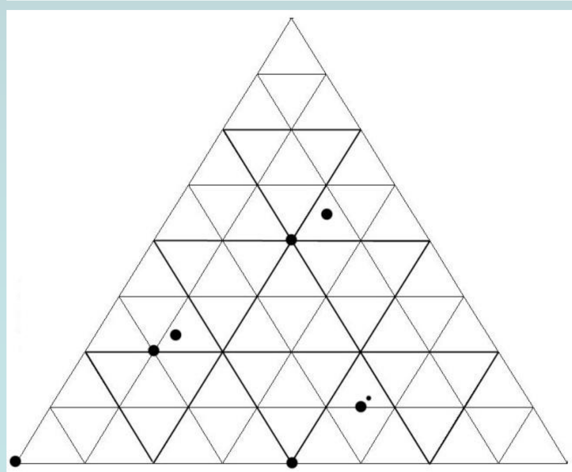


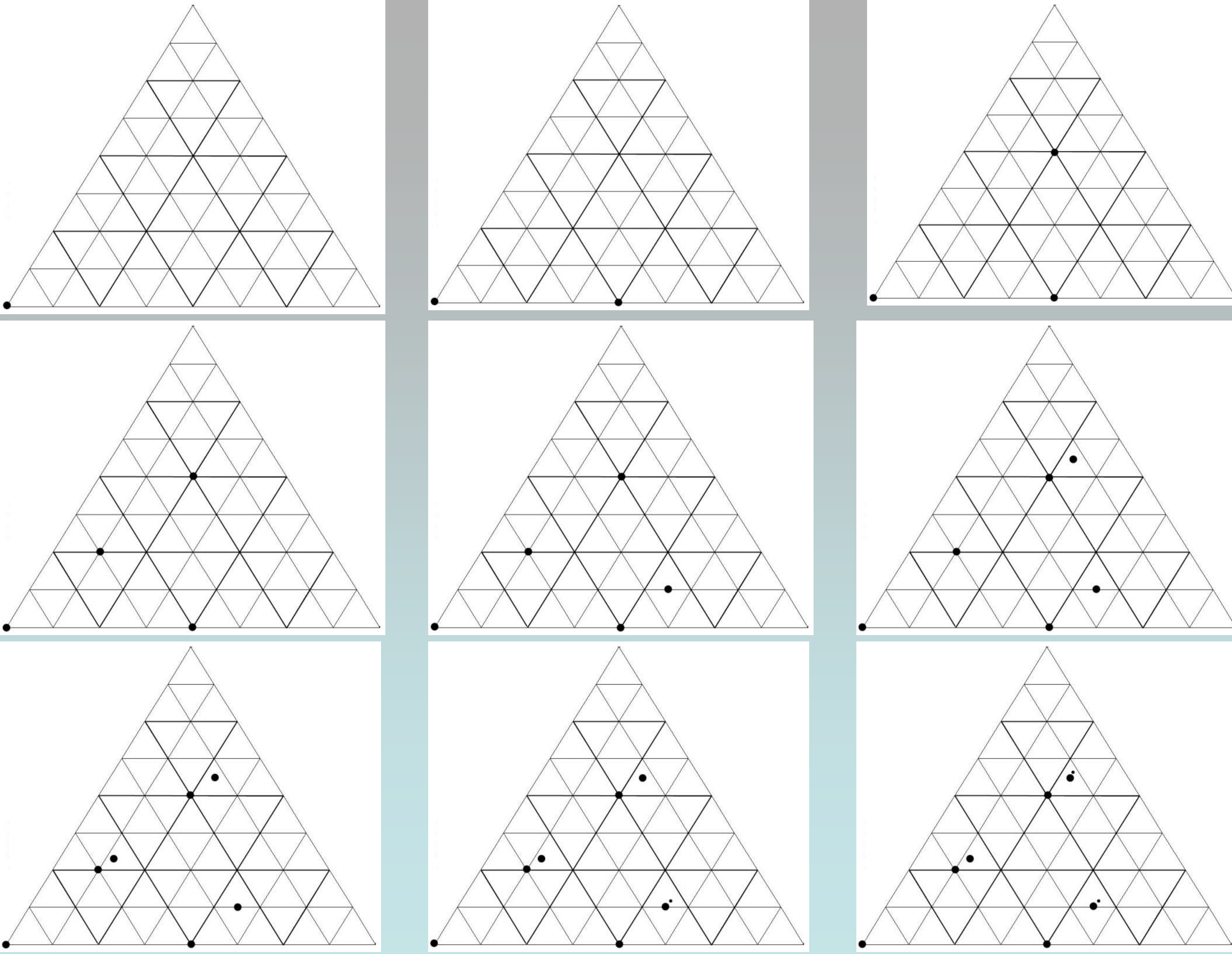


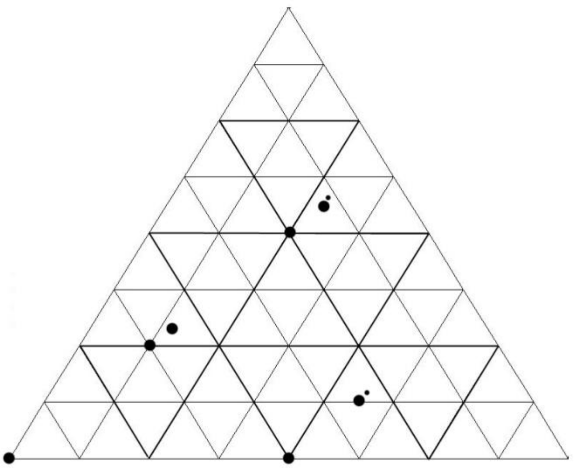


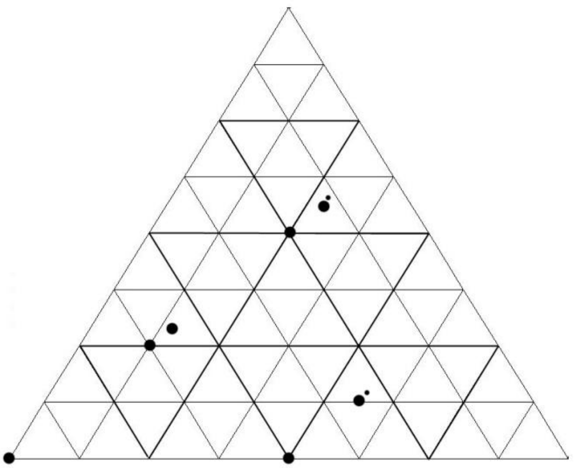


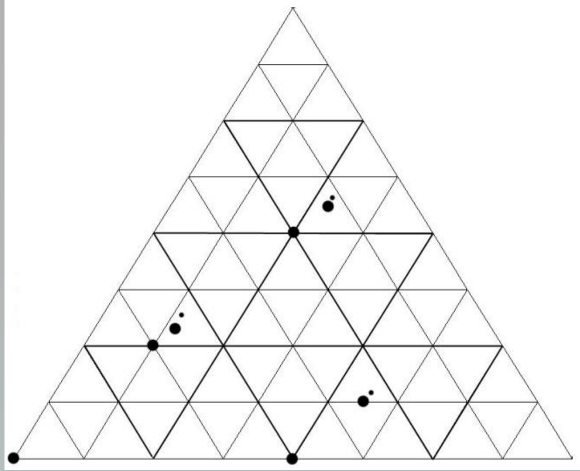


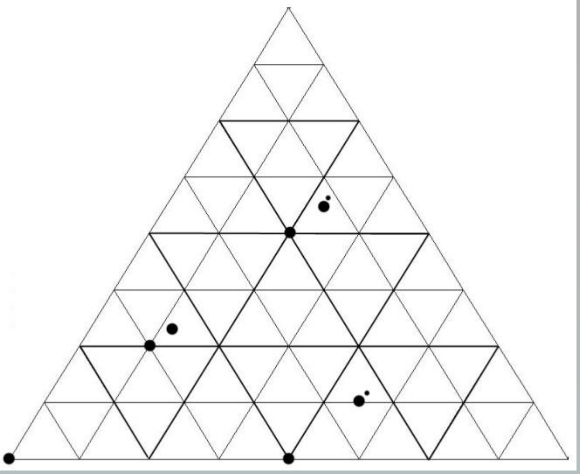
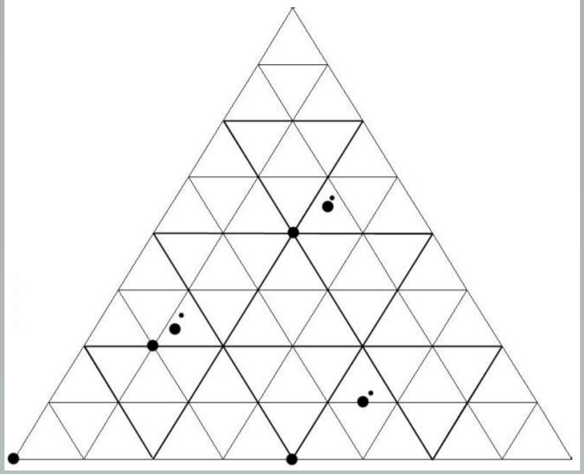
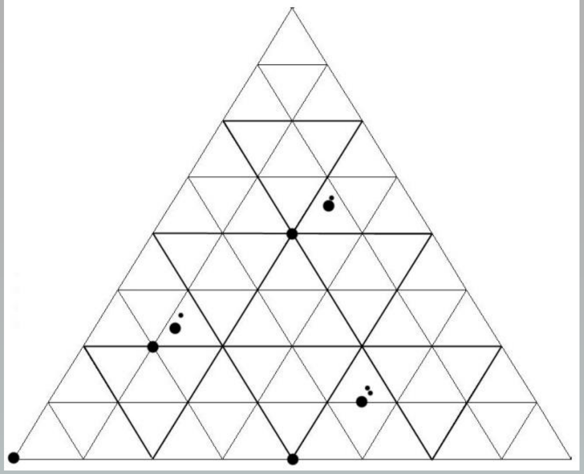









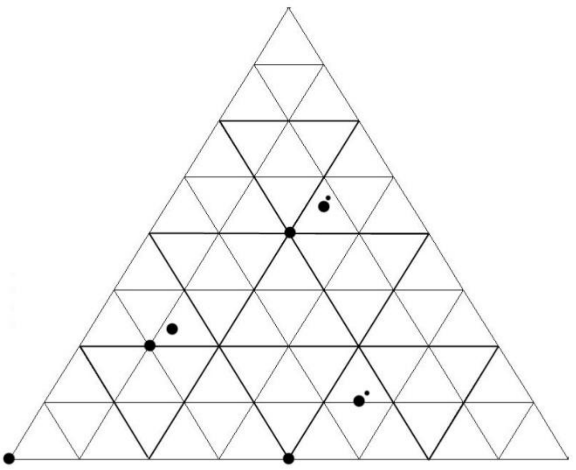
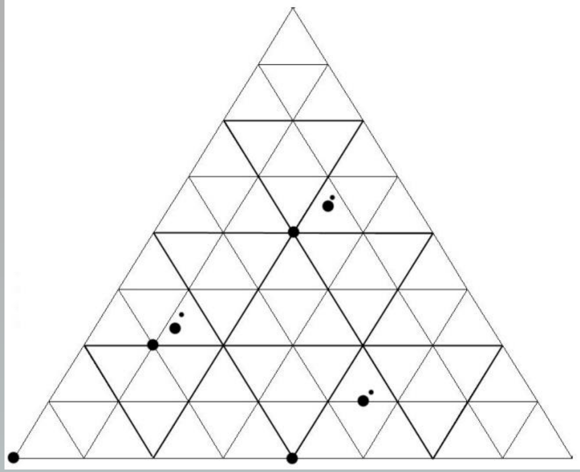
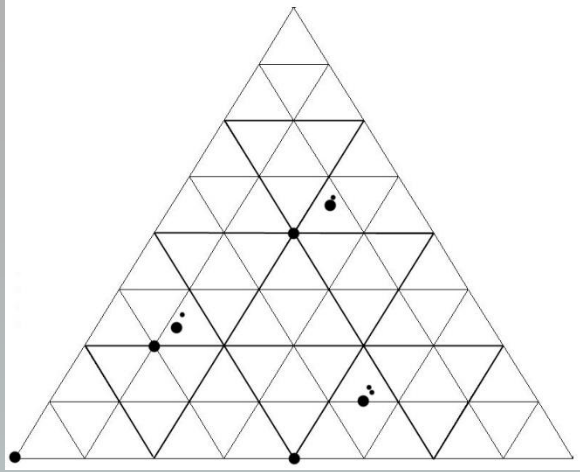
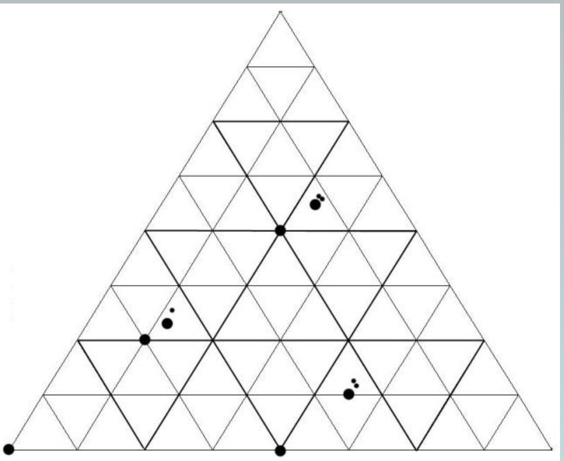


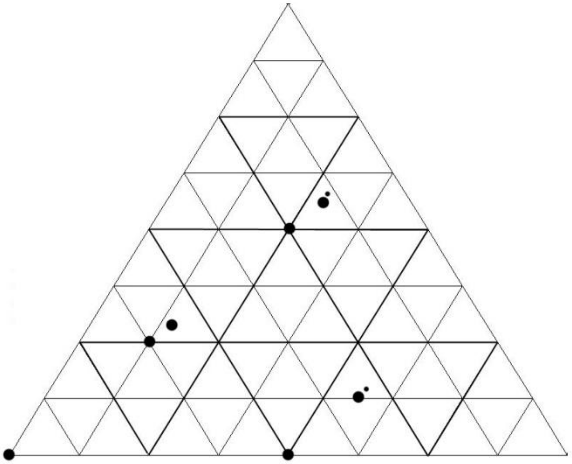
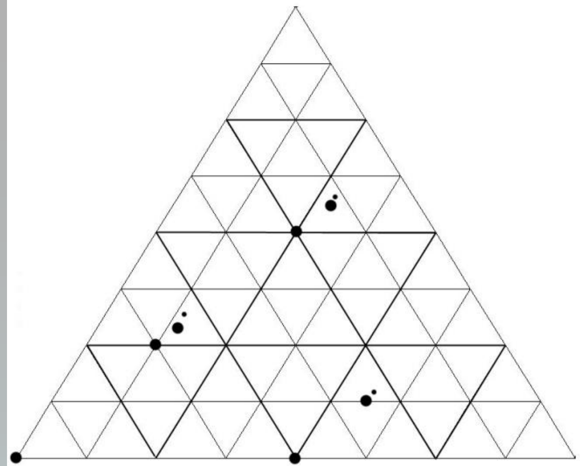
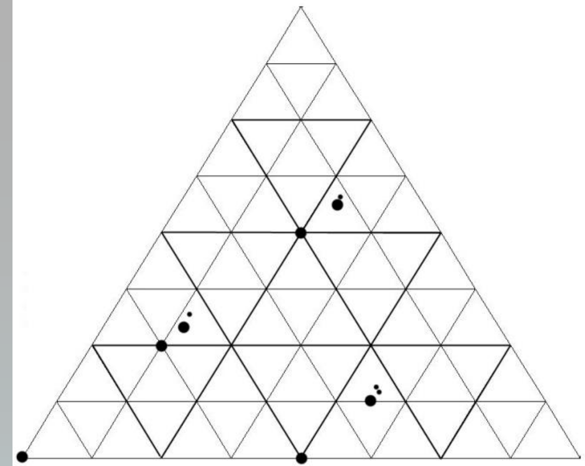
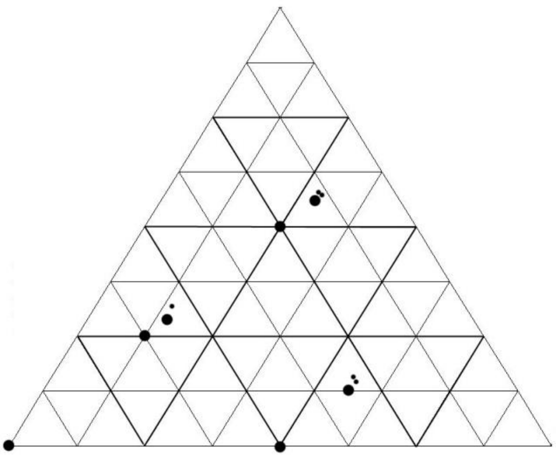
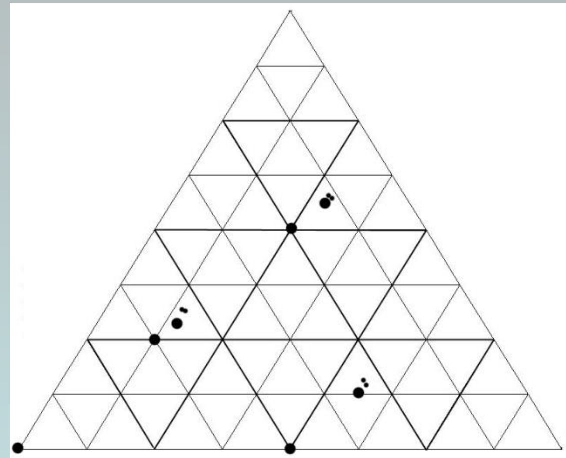


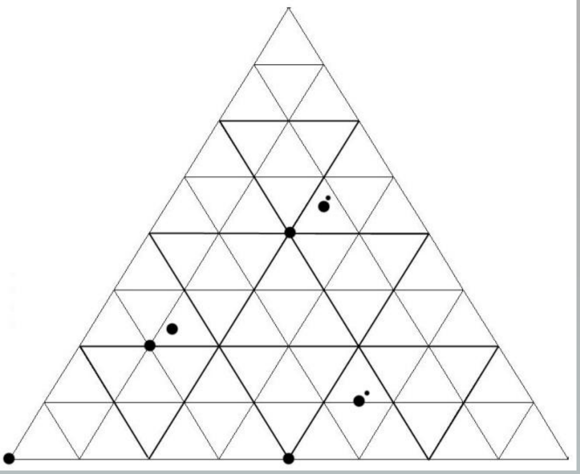
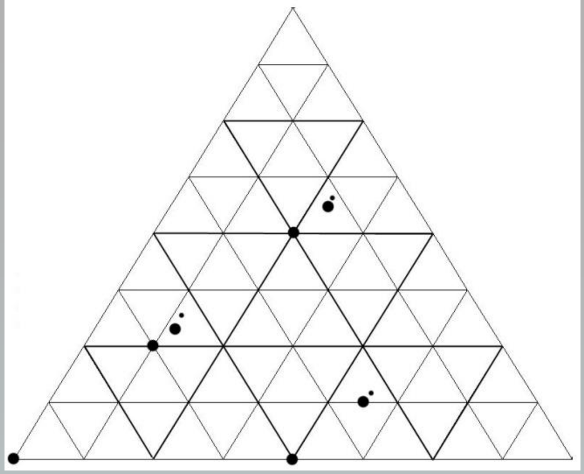
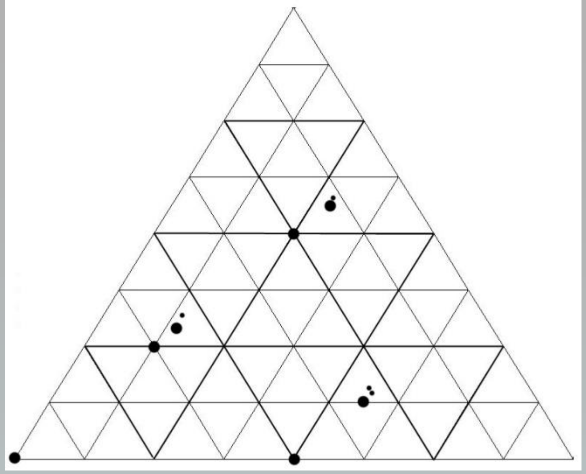
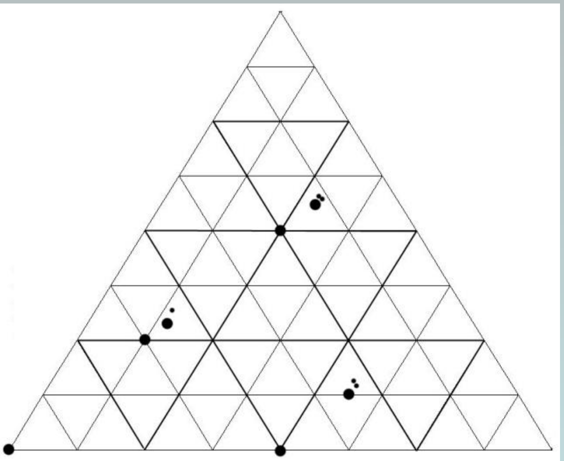
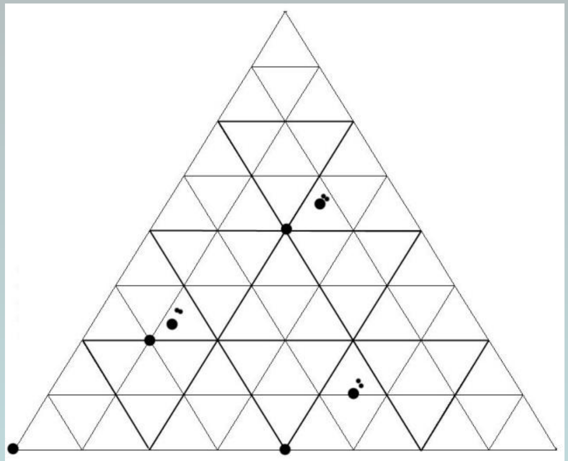
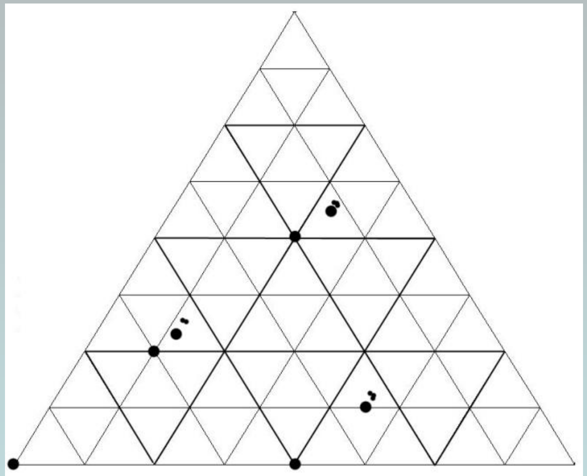


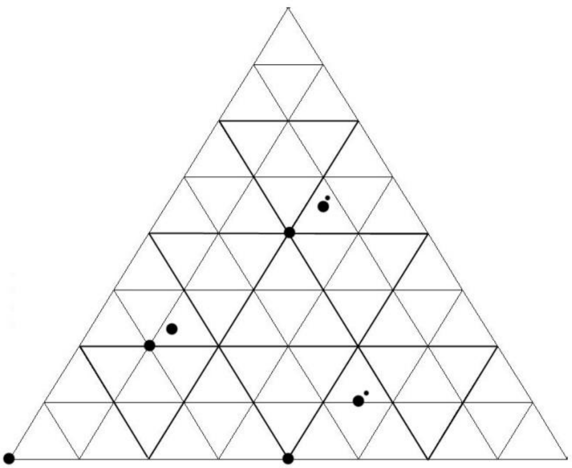
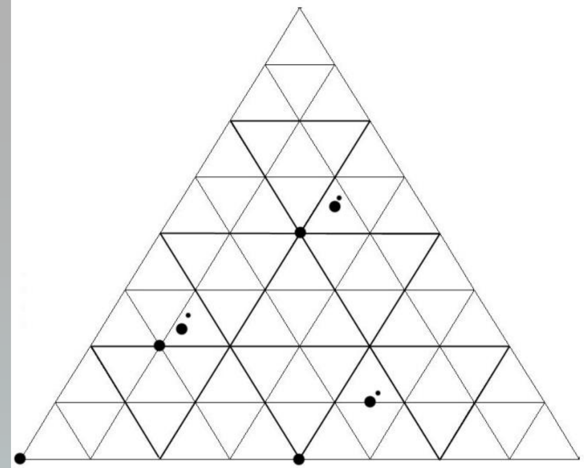
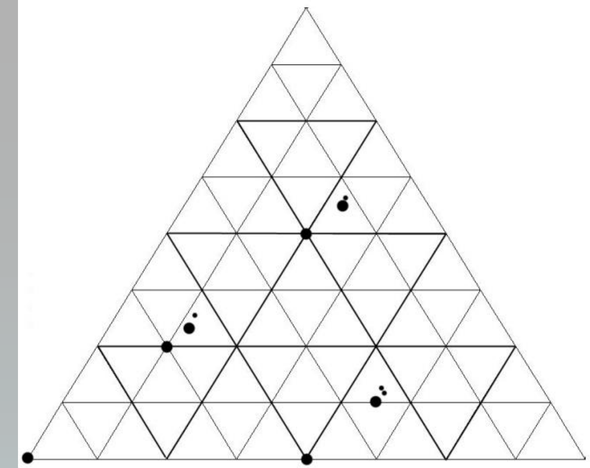
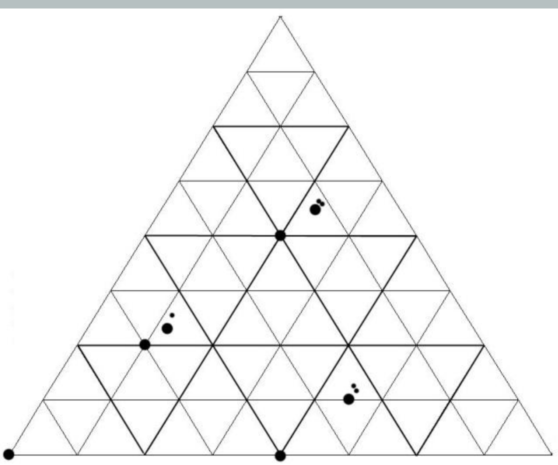
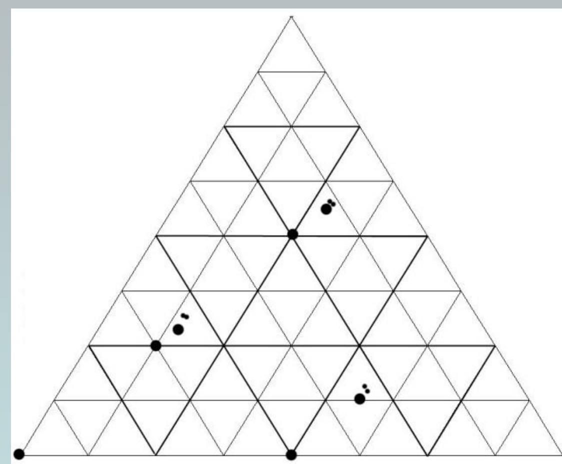
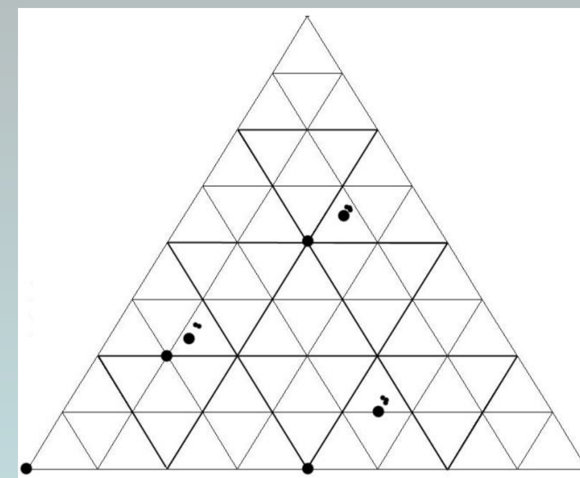
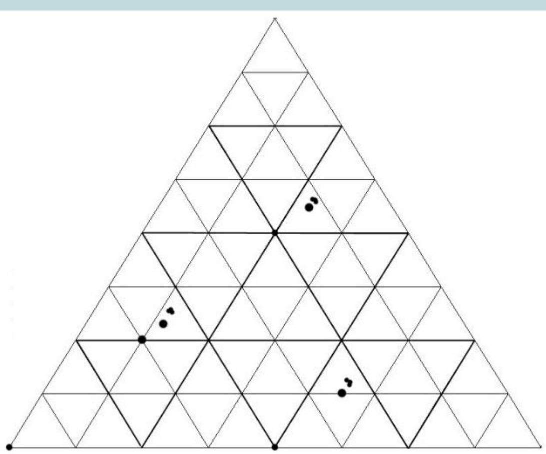


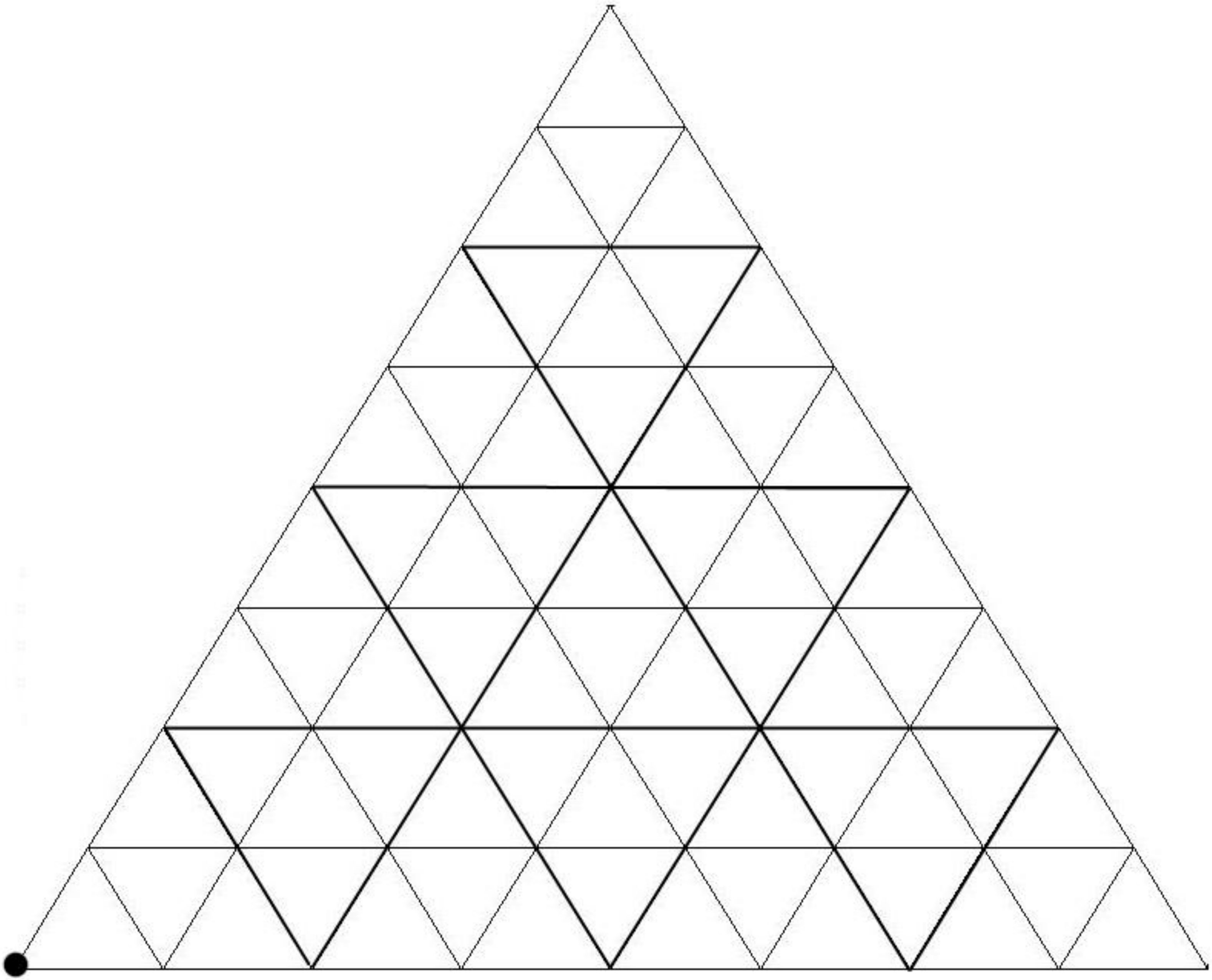




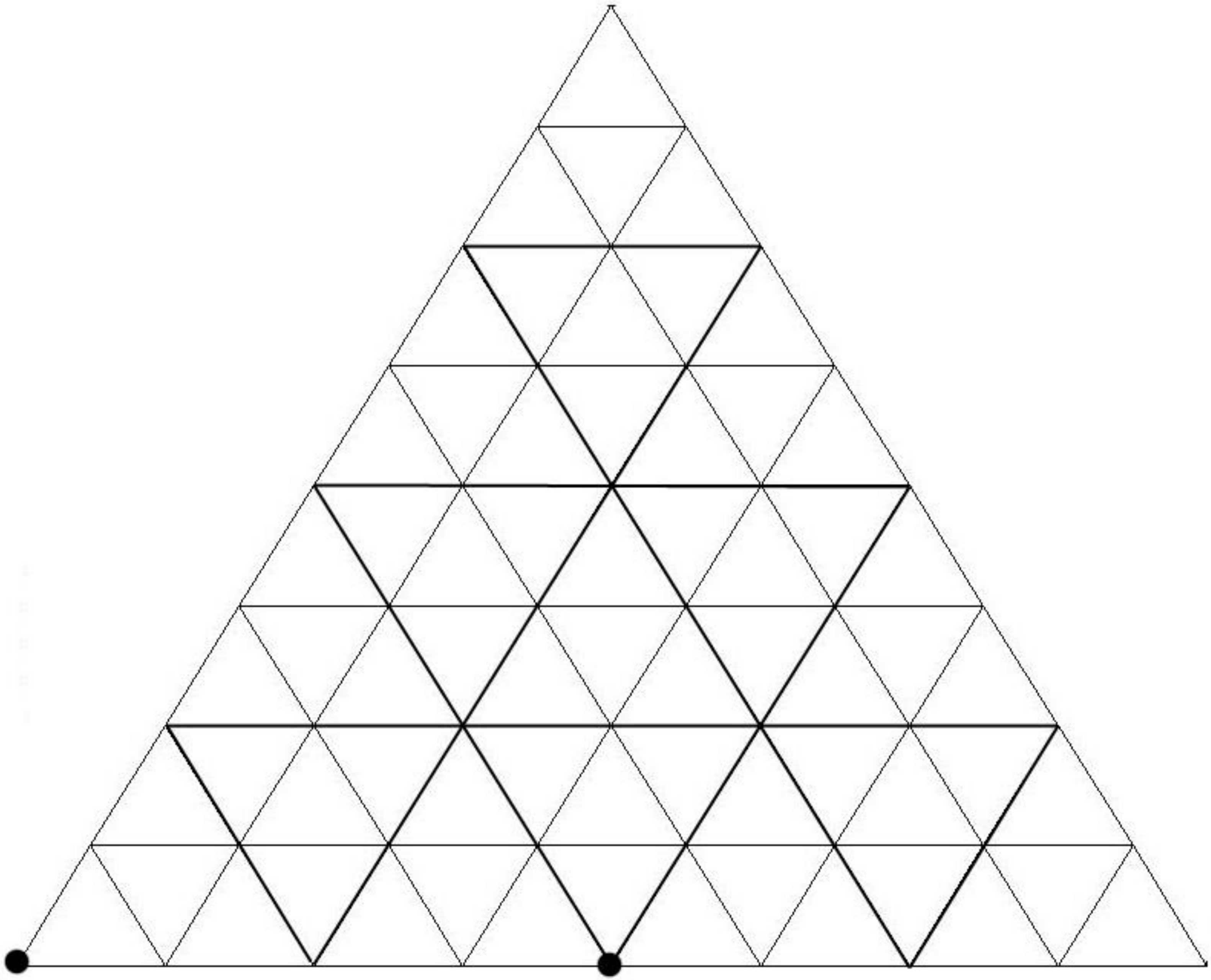





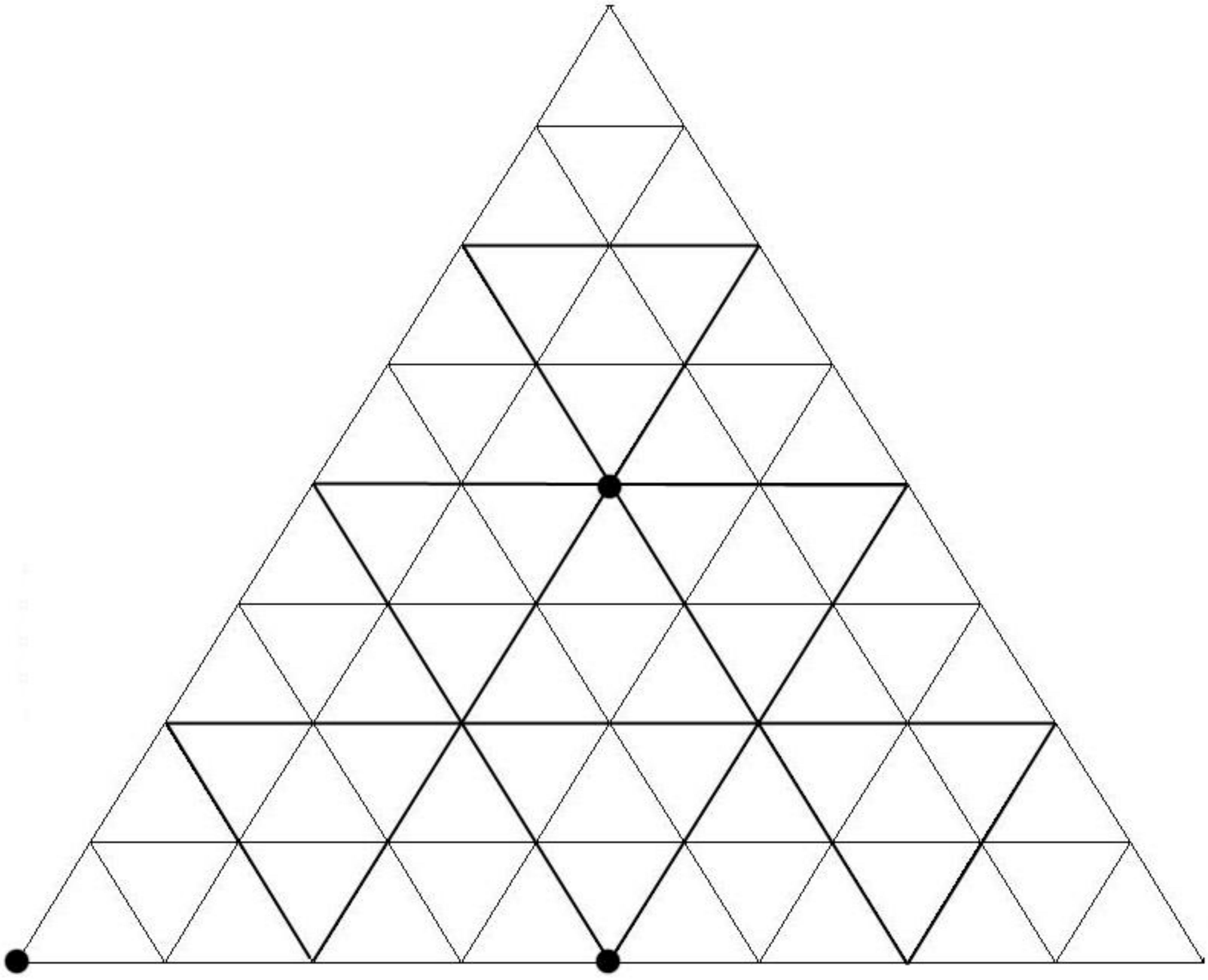






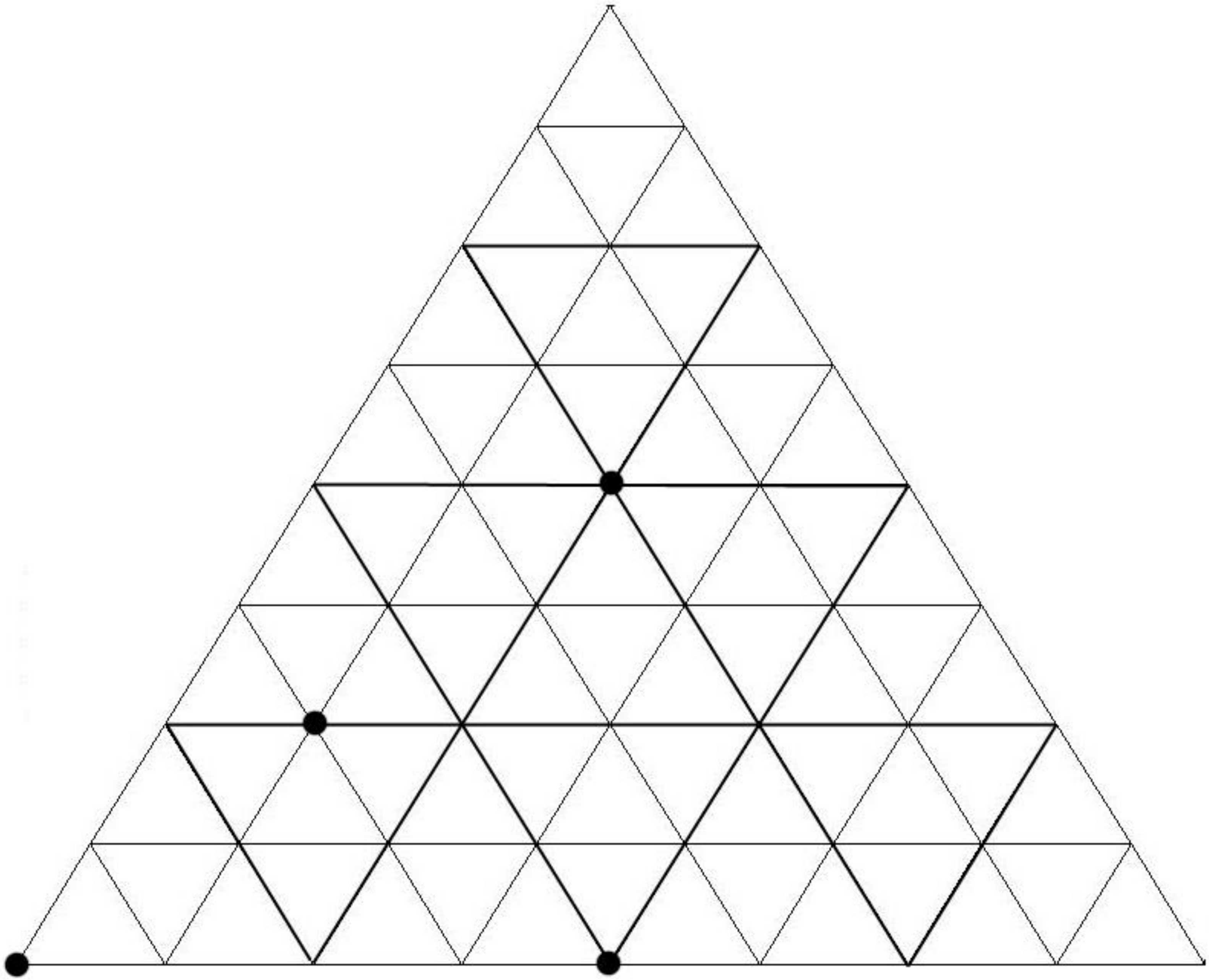







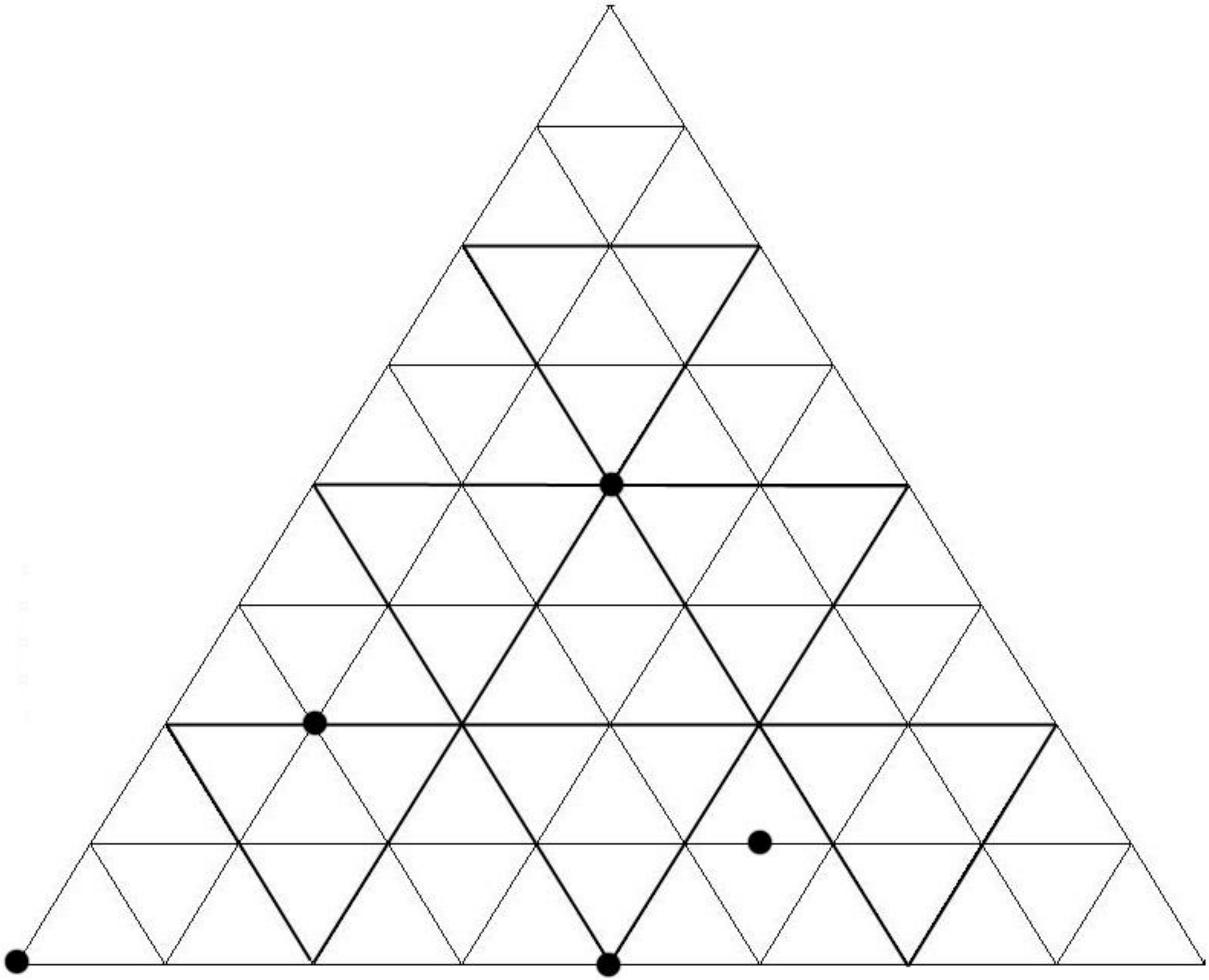








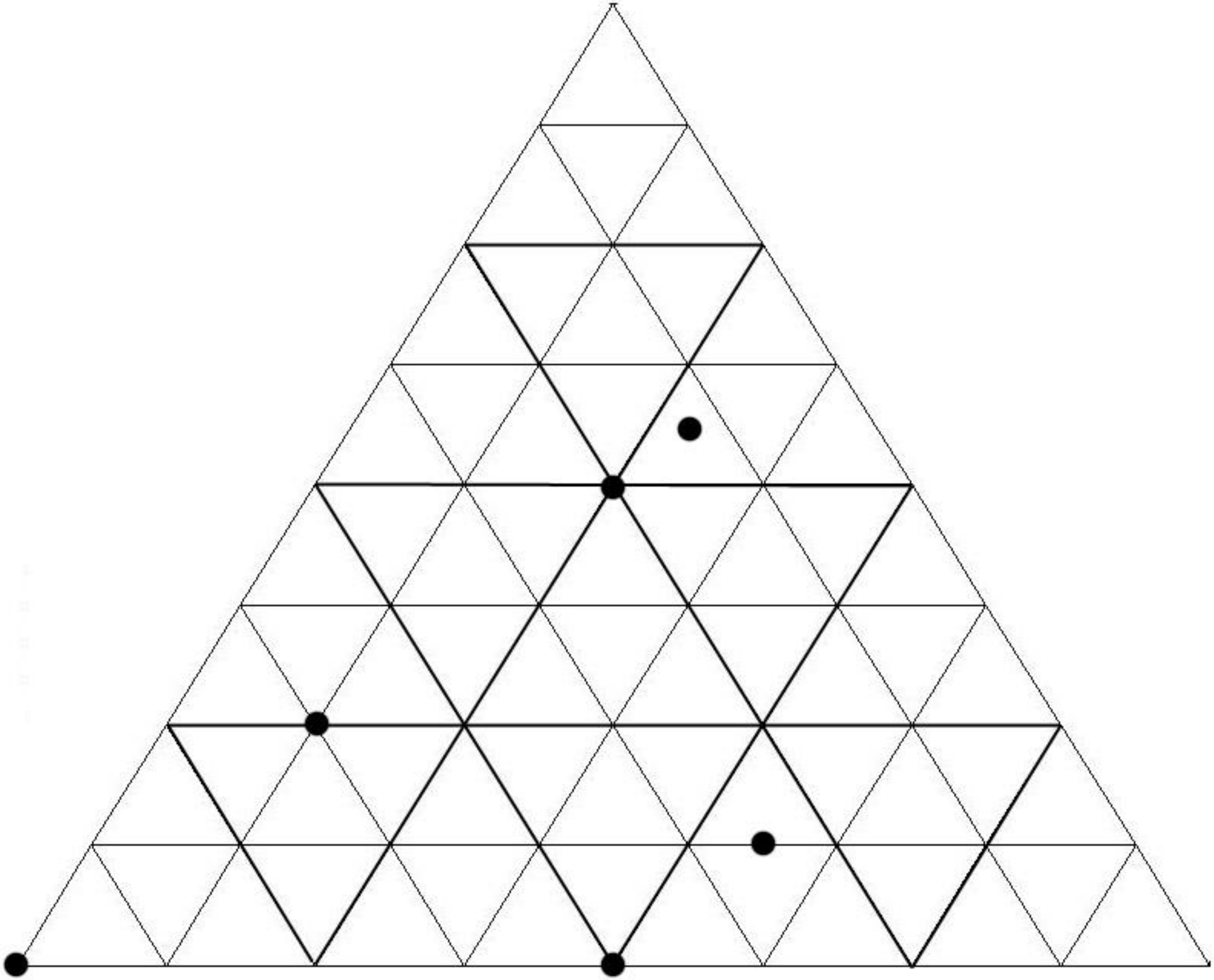


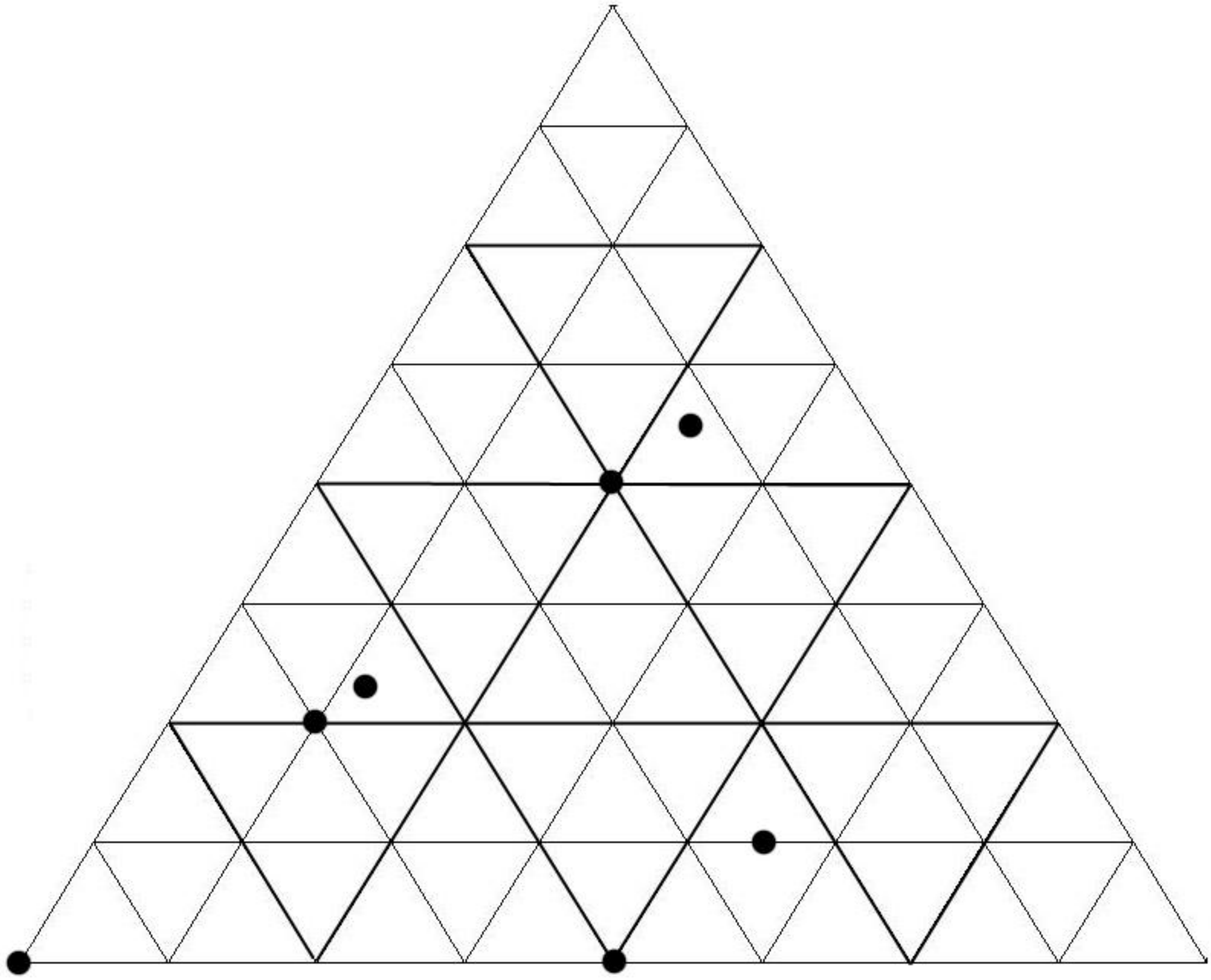


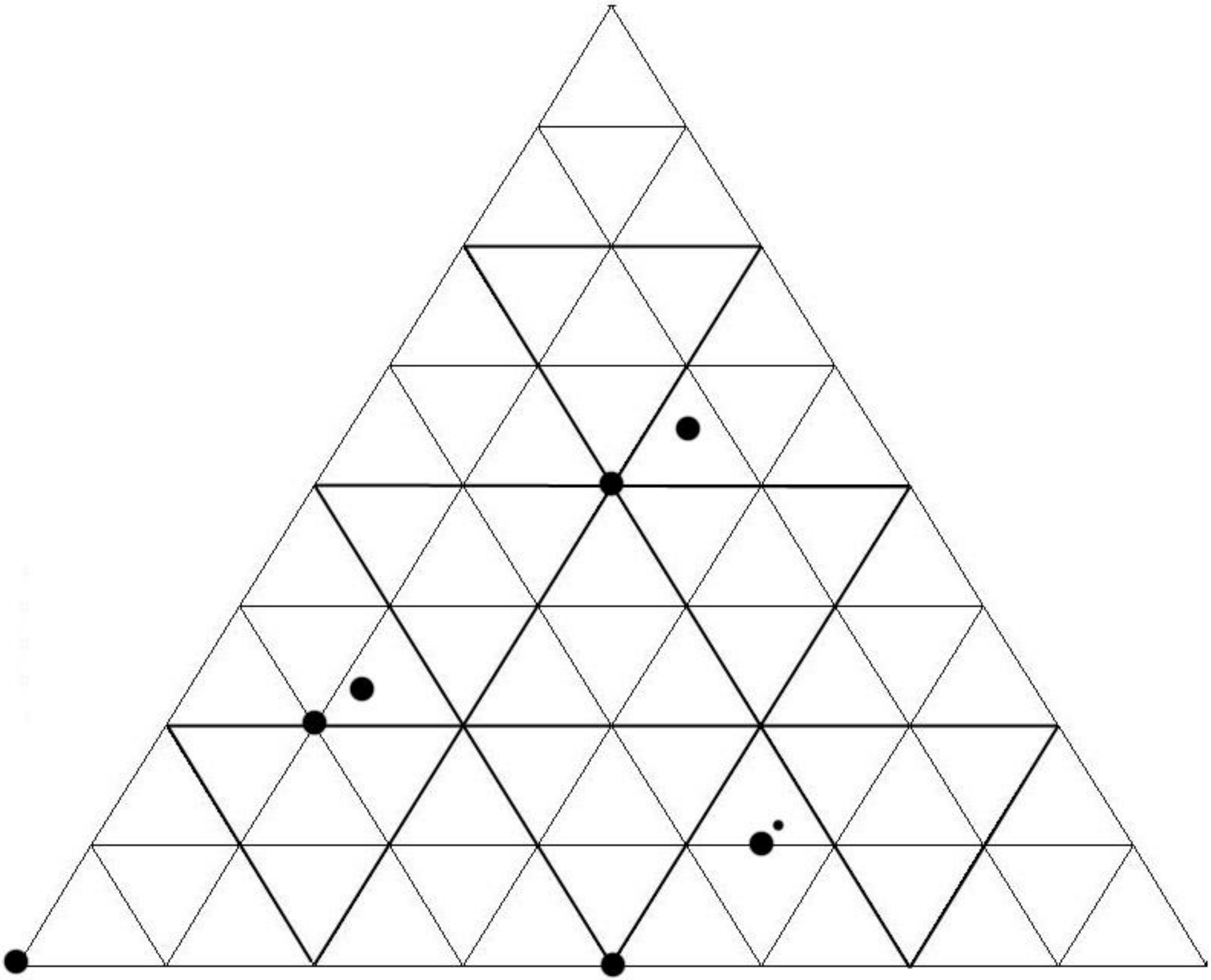


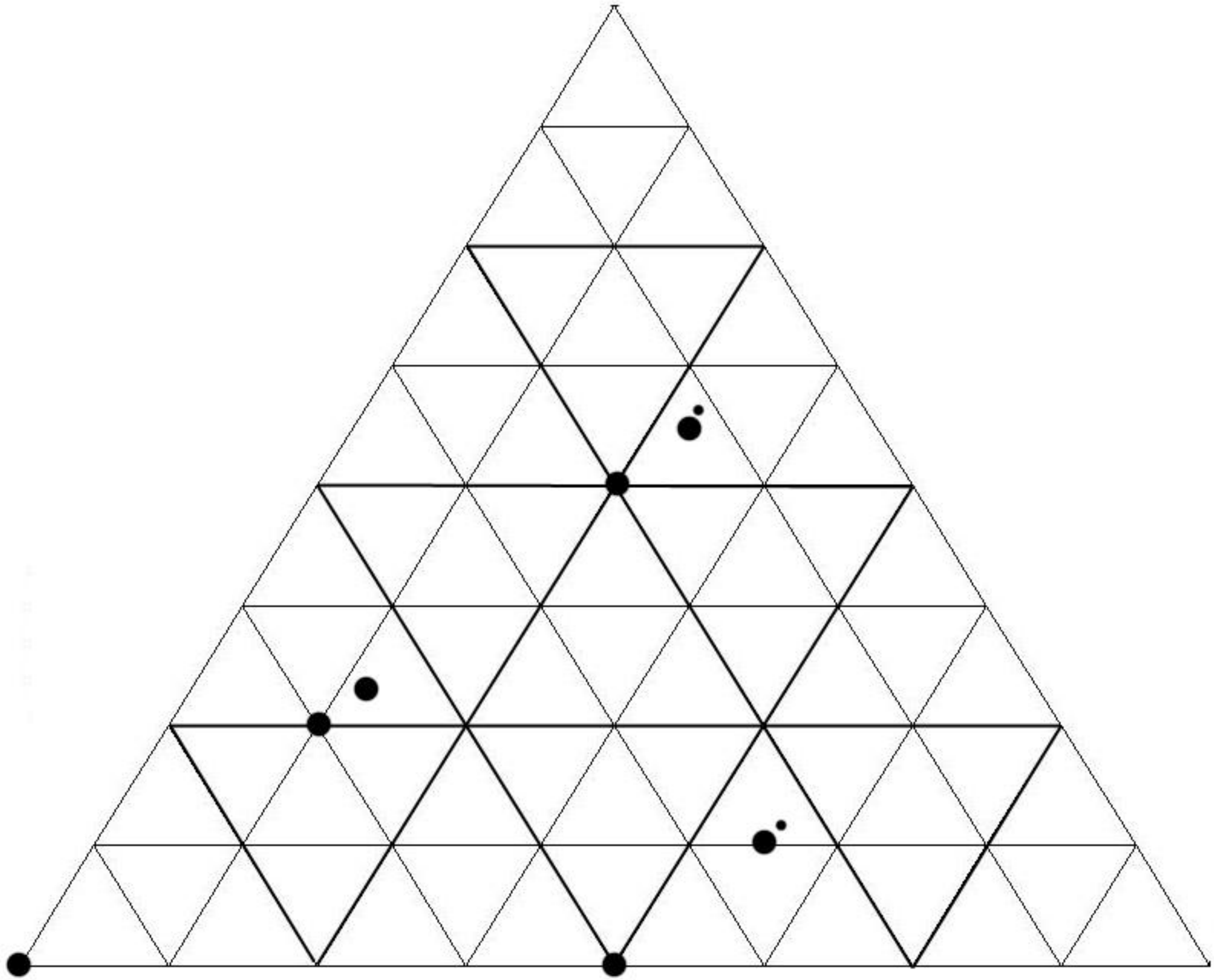


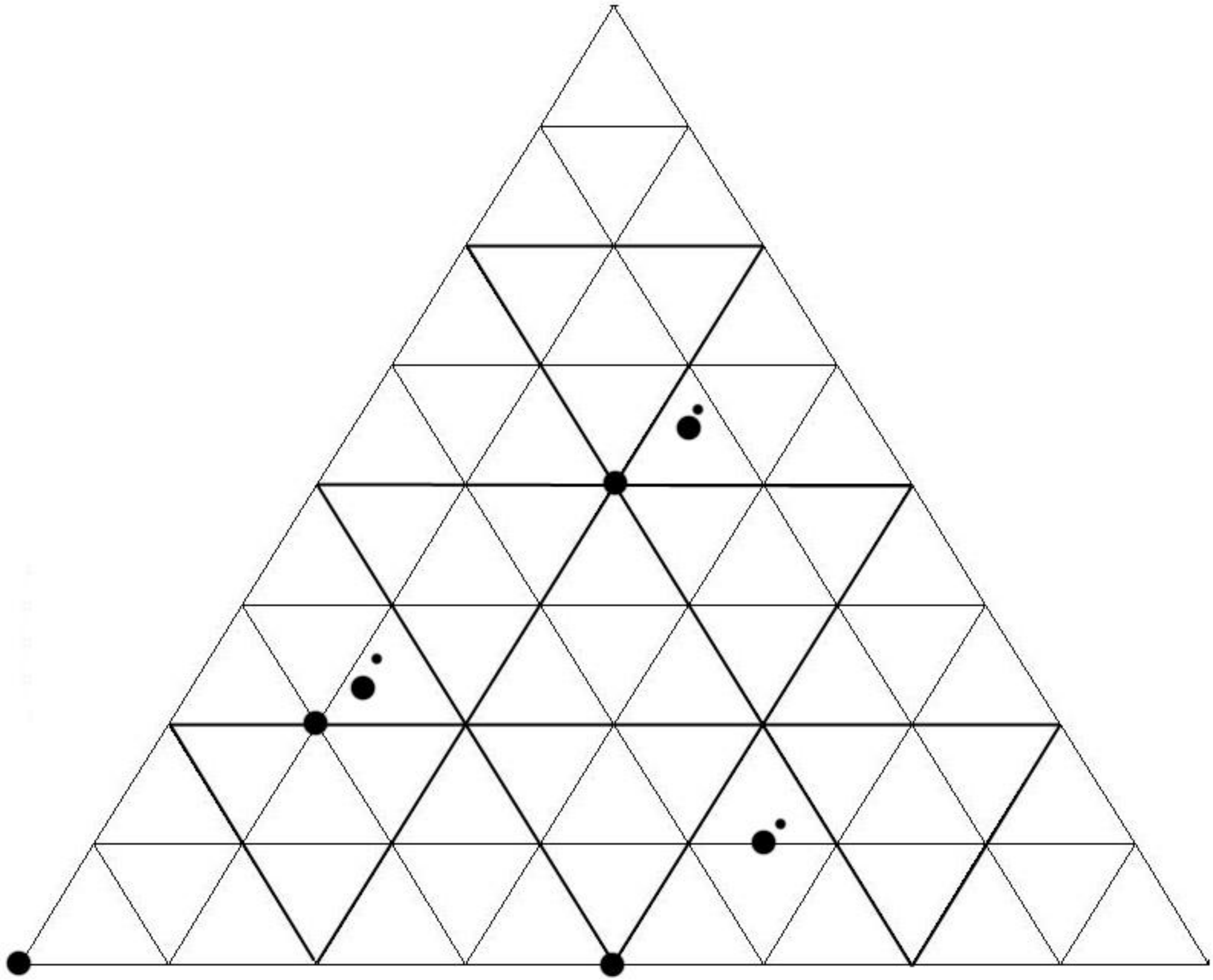


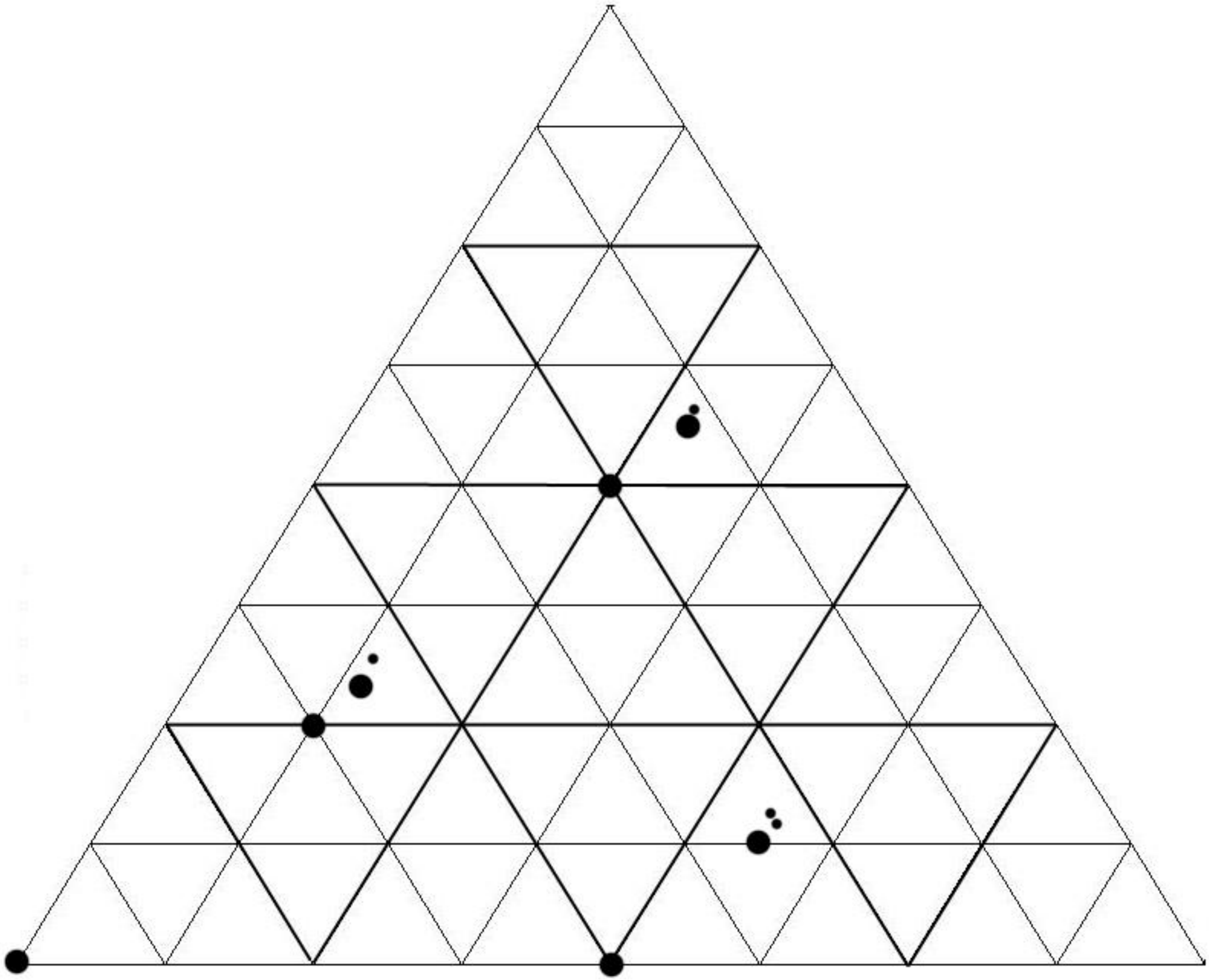


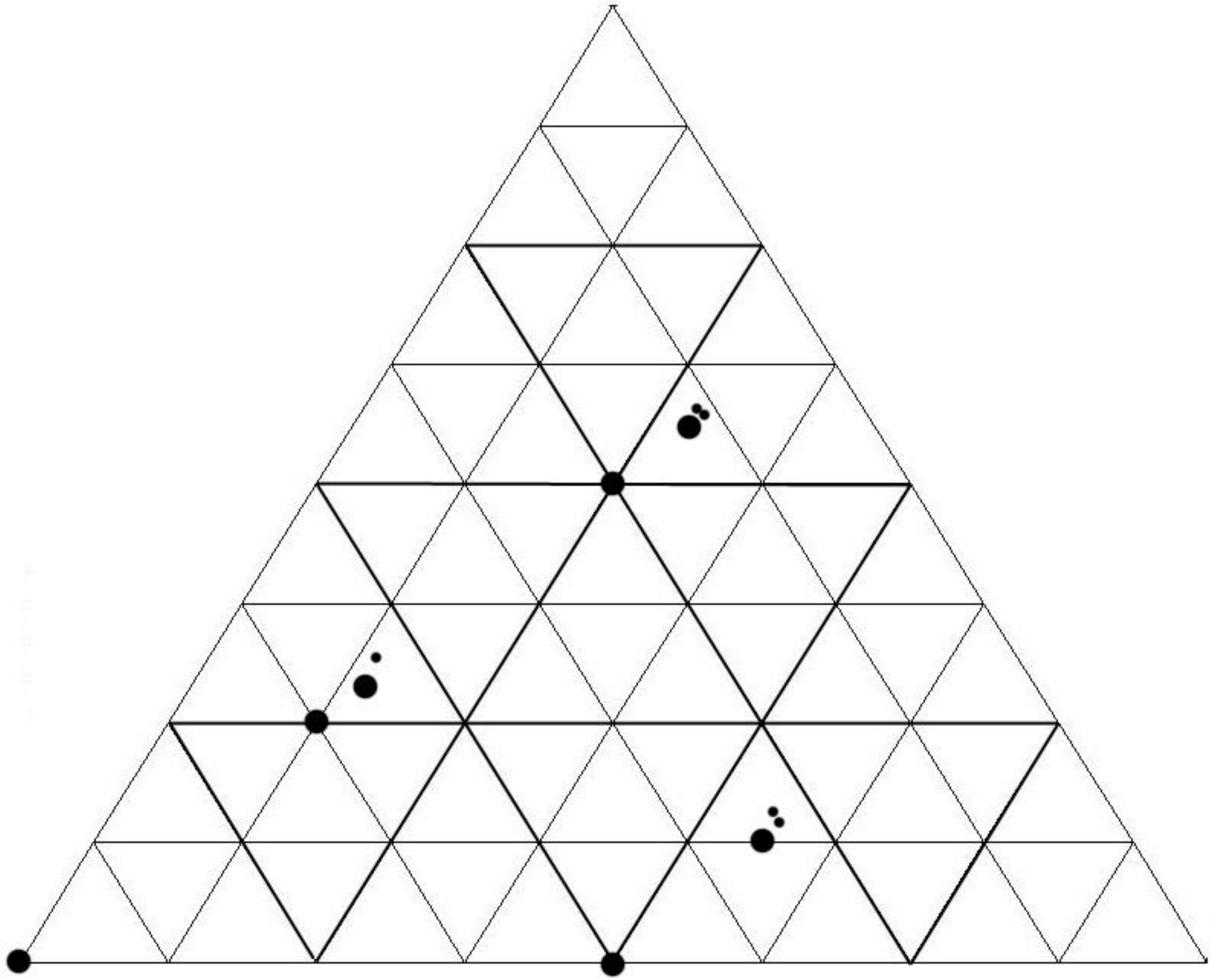


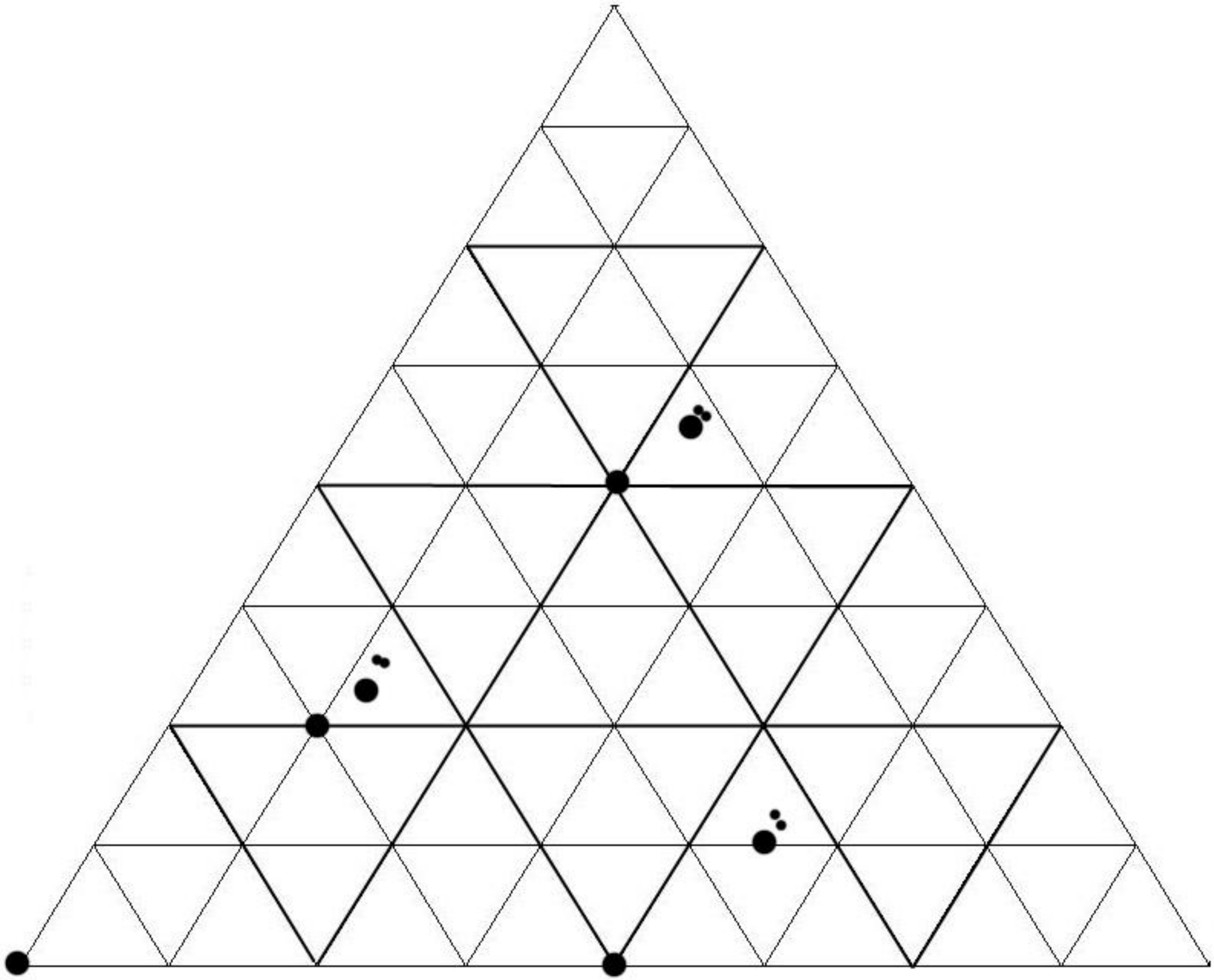


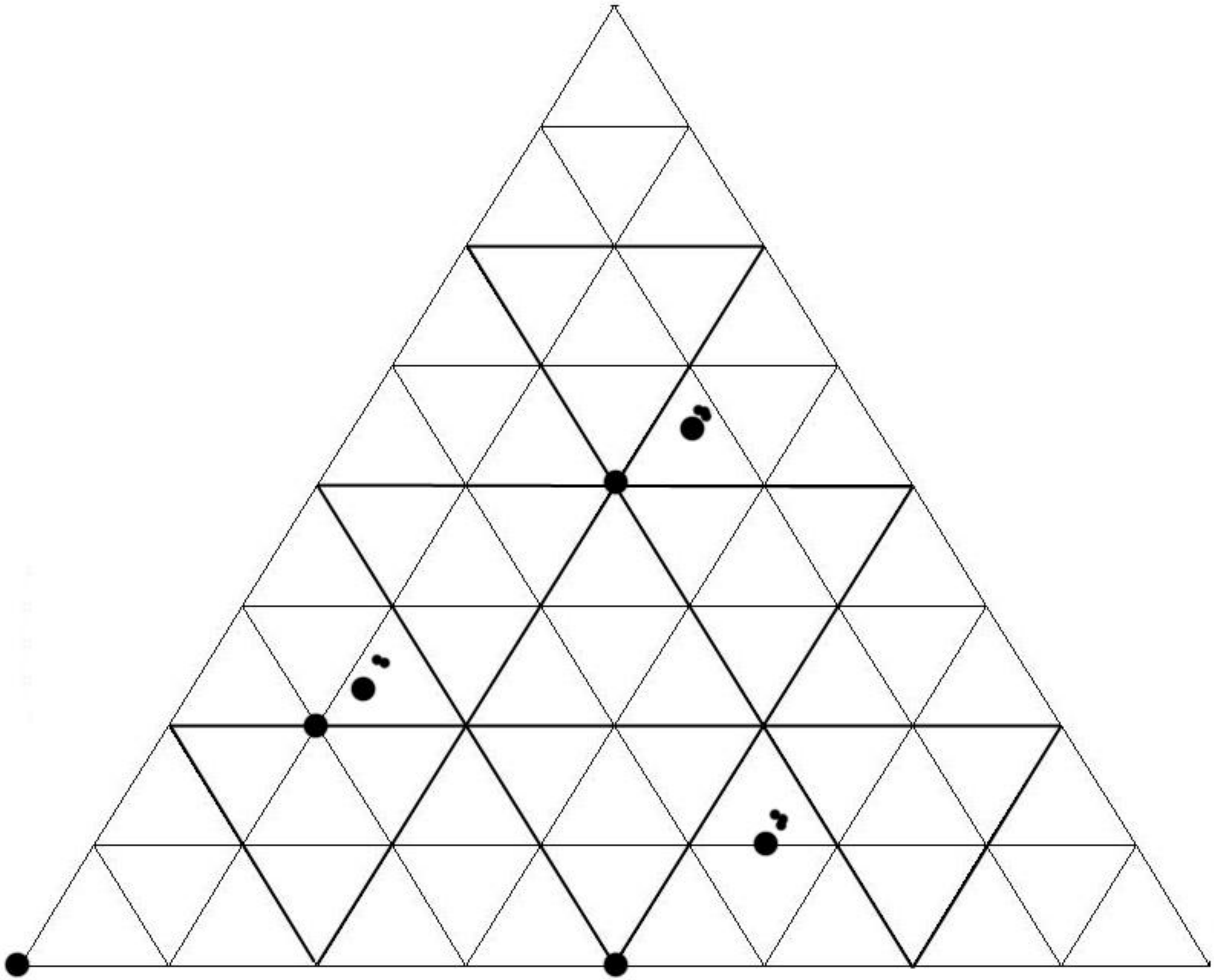


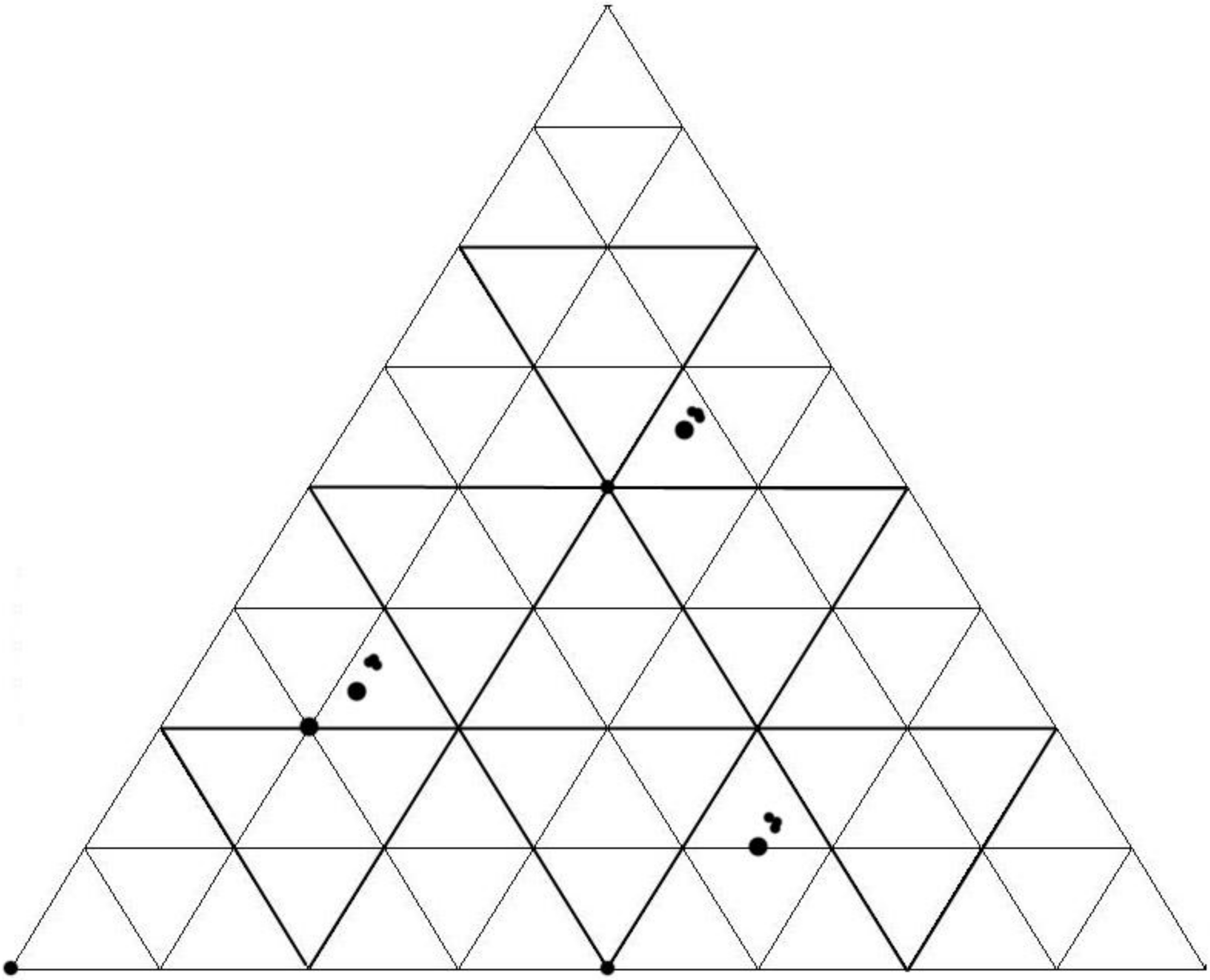






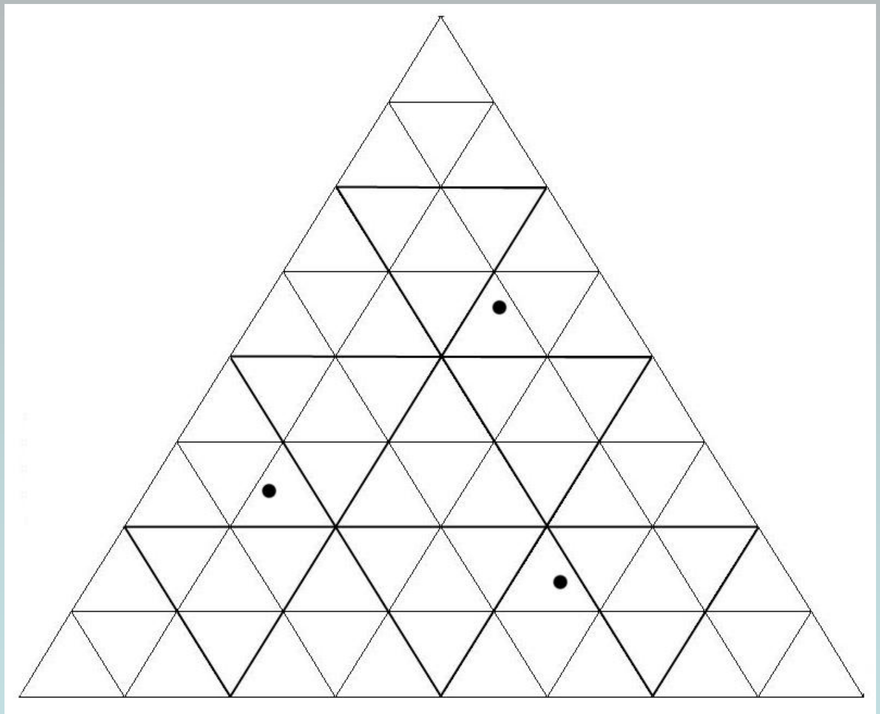




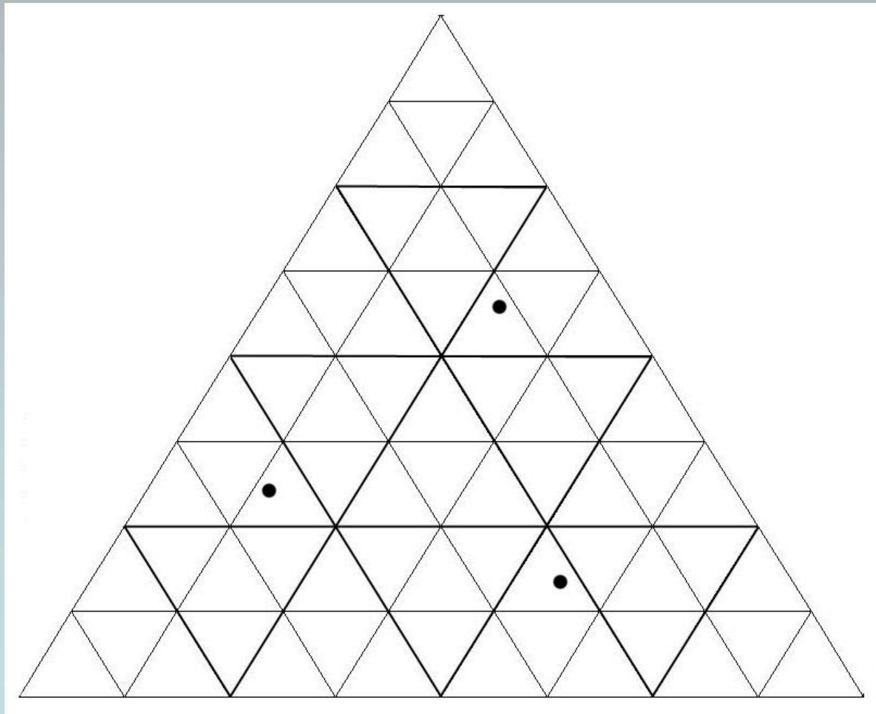


Looks like the game points are converging to 3 distinct points;

Looks like the game points are converging to 3 distinct points;

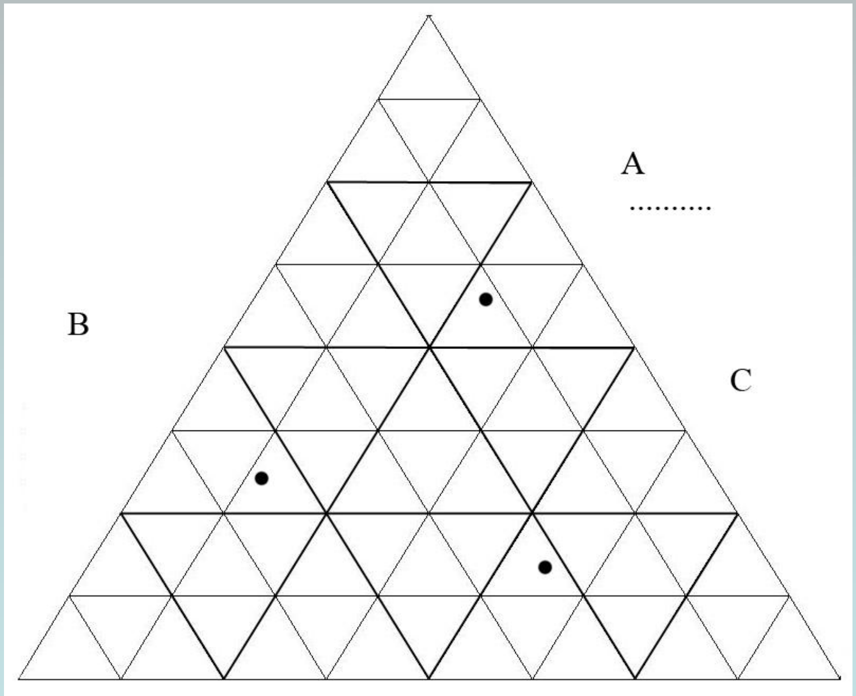


Looks like the game points are converging to 3 distinct points;

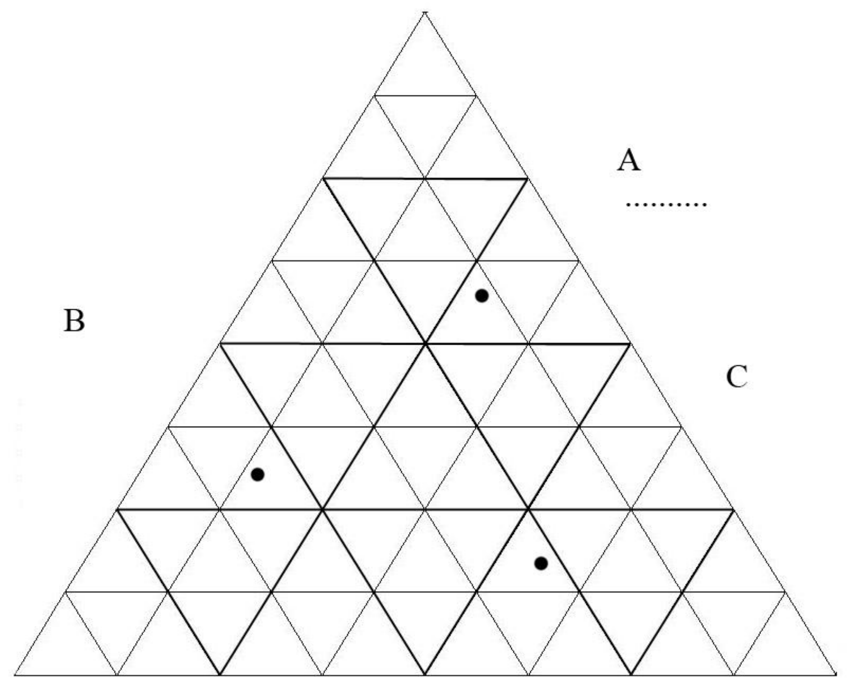


Why ?

Call these 3 points A, B, and C. Let •••••••• denote the address of point A.

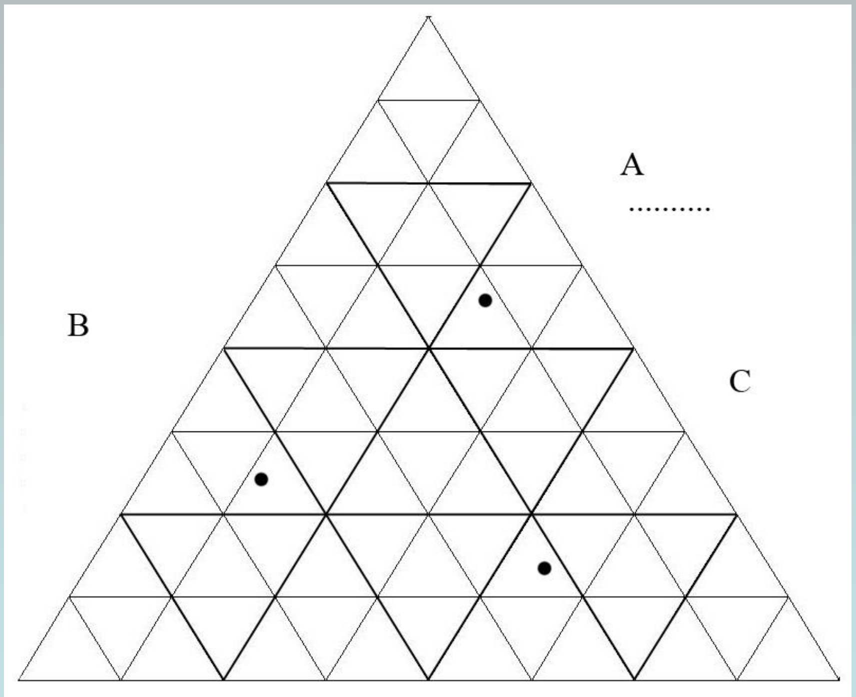


Call these 3 points A, B, and C. Let •••••••• denote the address of point A.



Now play the game with the game numbers 123123123.......

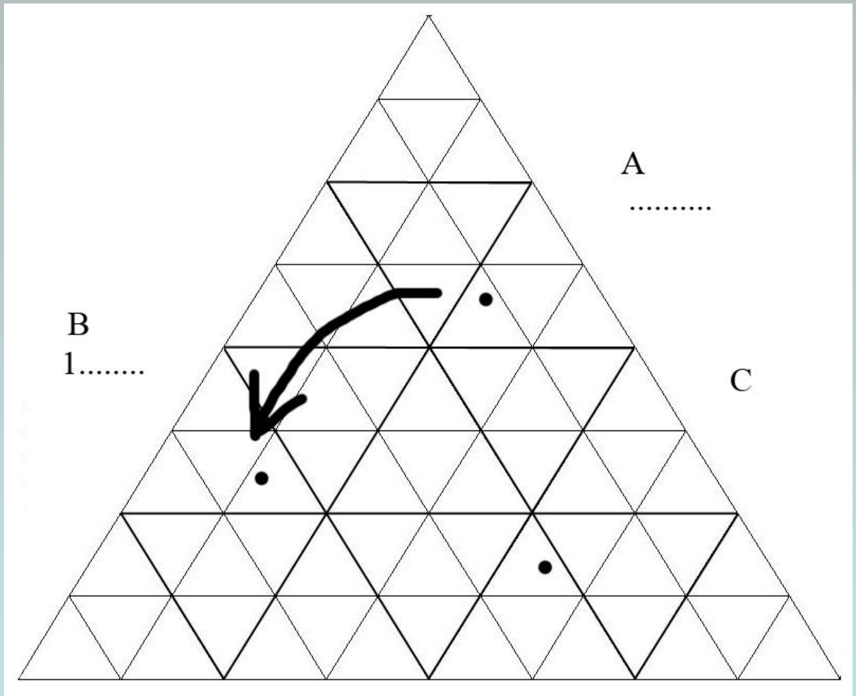
Call these 3 points A, B, and C. Let •••••••• denote the address of point A.



Now play the game with the game numbers 123123123.......

Address of point B is

Call these 3 points A, B, and C. Let •••••••• denote the address of point A.

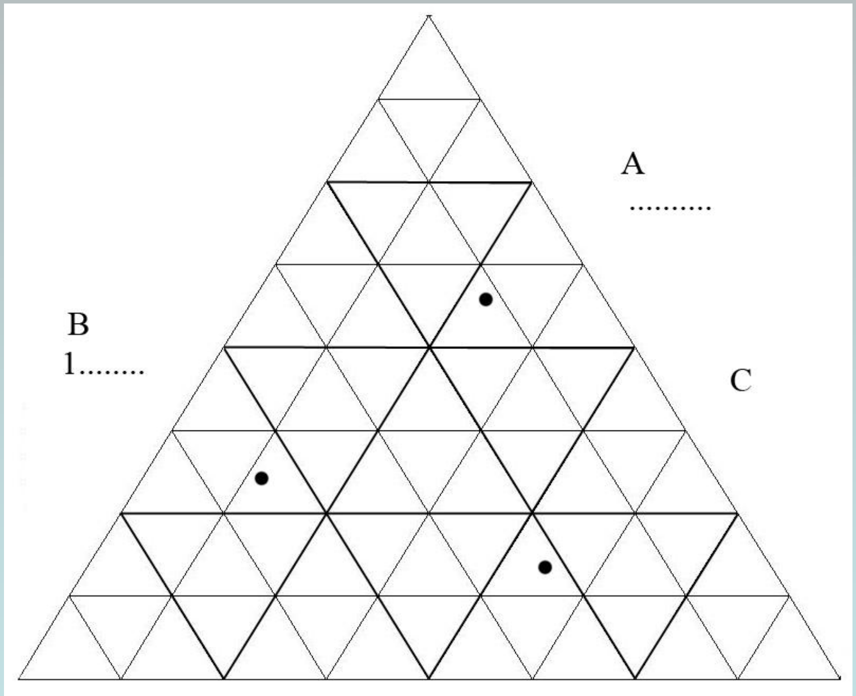


Now play the game with the game numbers 123123123.......

Address of point B is

1.........

Call these 3 points A, B, and C. Let •••••••• denote the address of point A.



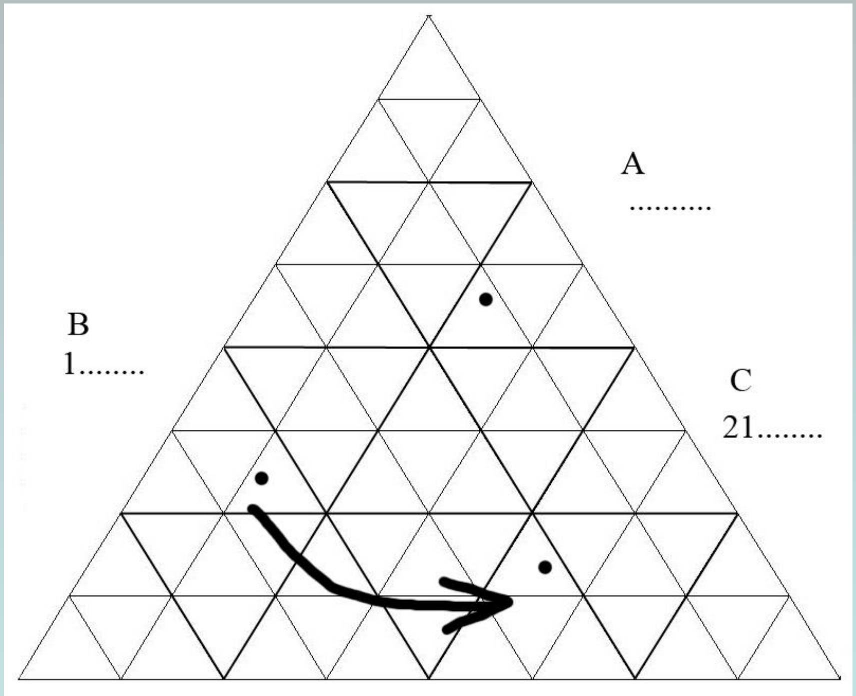
Now play the game with the game numbers 123123123.......

Address of point B is

1.........

Address of point C is

Call these 3 points A, B, and C. Let •••••••• denote the address of point A.

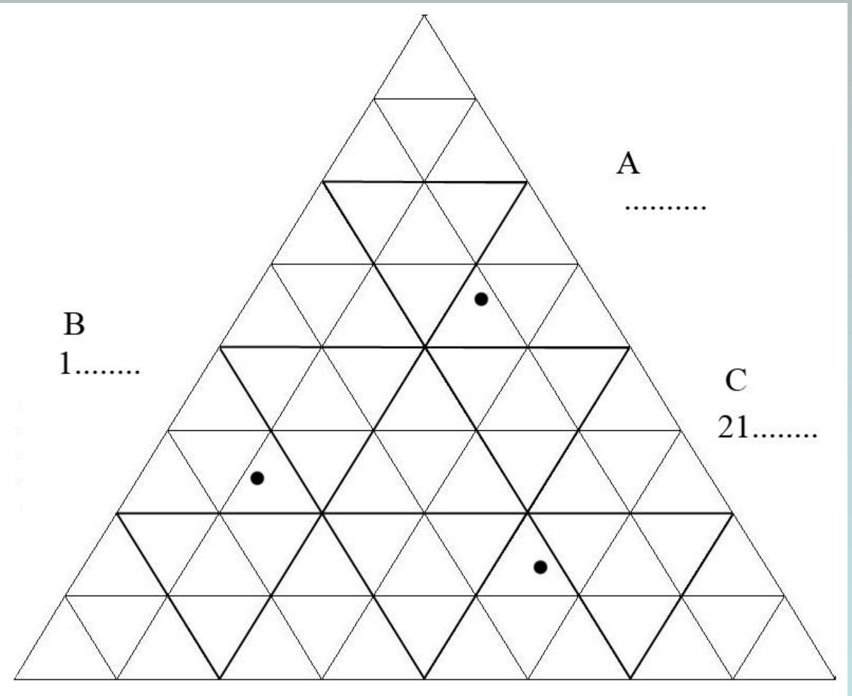


Now play the game with the game numbers 123123123.......

Address of point B is 1.........

Address of point C is 21.........

Call these 3 points A, B, and C. Let •••••••• denote the address of point A.



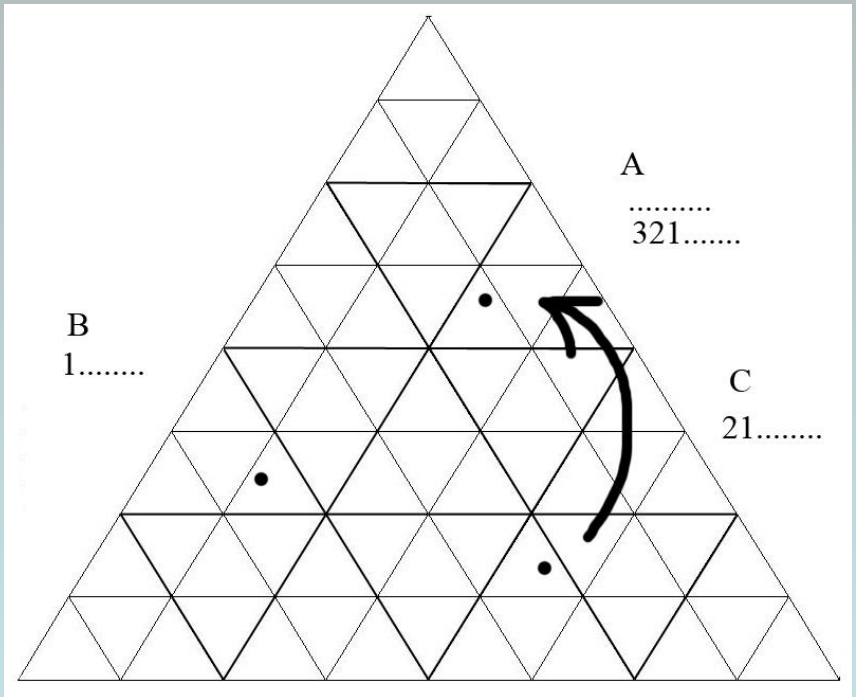
Now play the game with the game numbers 123123123.......

Address of point B is 1.........

Address of point C is 21.........

Address of point A then is

Call these 3 points A, B, and C. Let •••••••• denote the address of point A.



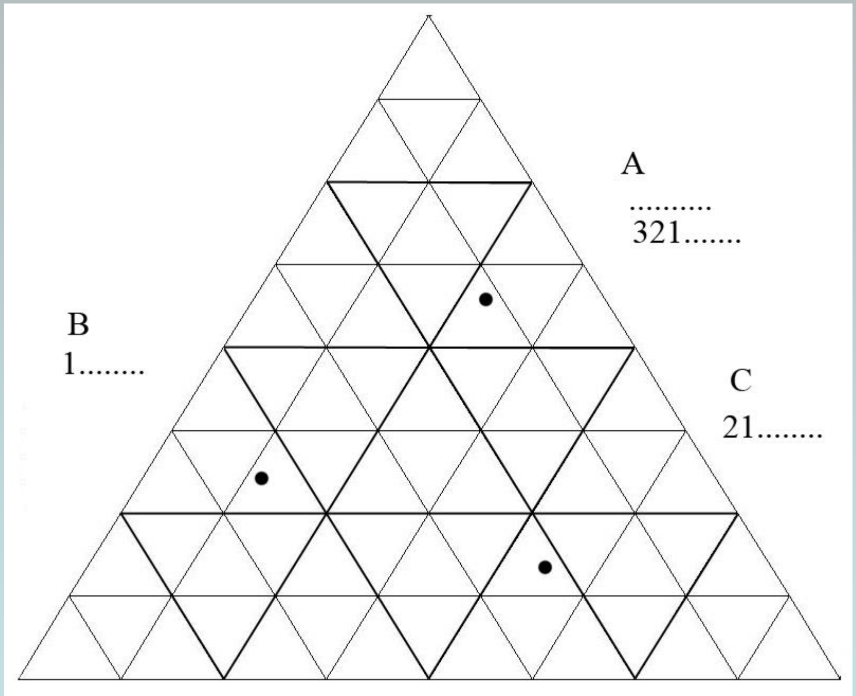
Now play the game with the game numbers 123123123.......

Address of point B is 1.........

Address of point C is 21.........

Address of point A then is 321........

Call these 3 points A, B, and C. Let •••••••• denote the address of point A.

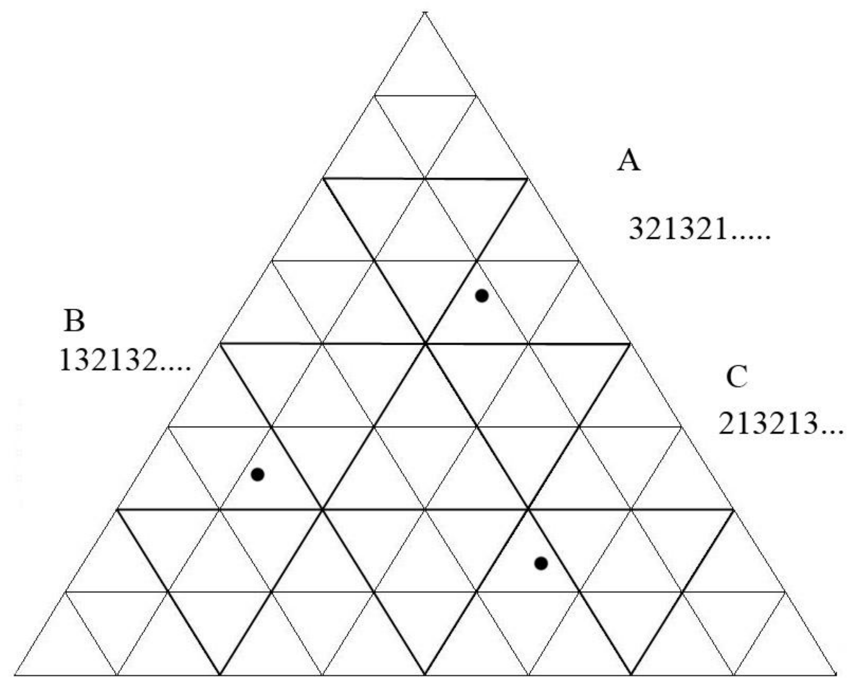


Now play the game with the game numbers 123123123.......

Address of point B is 1.........

Address of point C is 21.........

Address of point A then is 321....... = ...

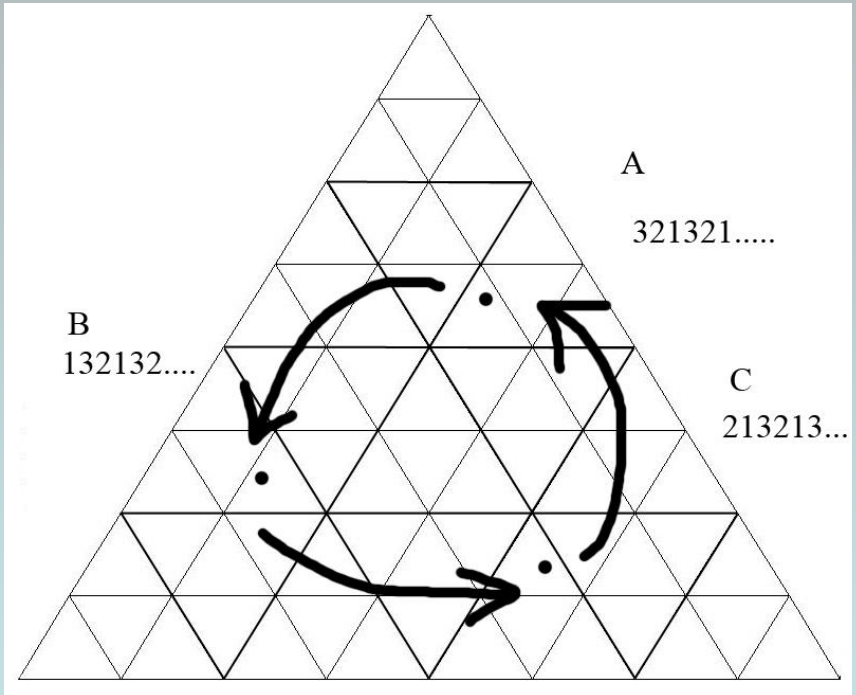


So the address of point A must be 321321....

And so address of B must be 132132.....

And the address of point C must be

213213.......



So the address of point A must be 321321....

And so address of B must be 132132.....

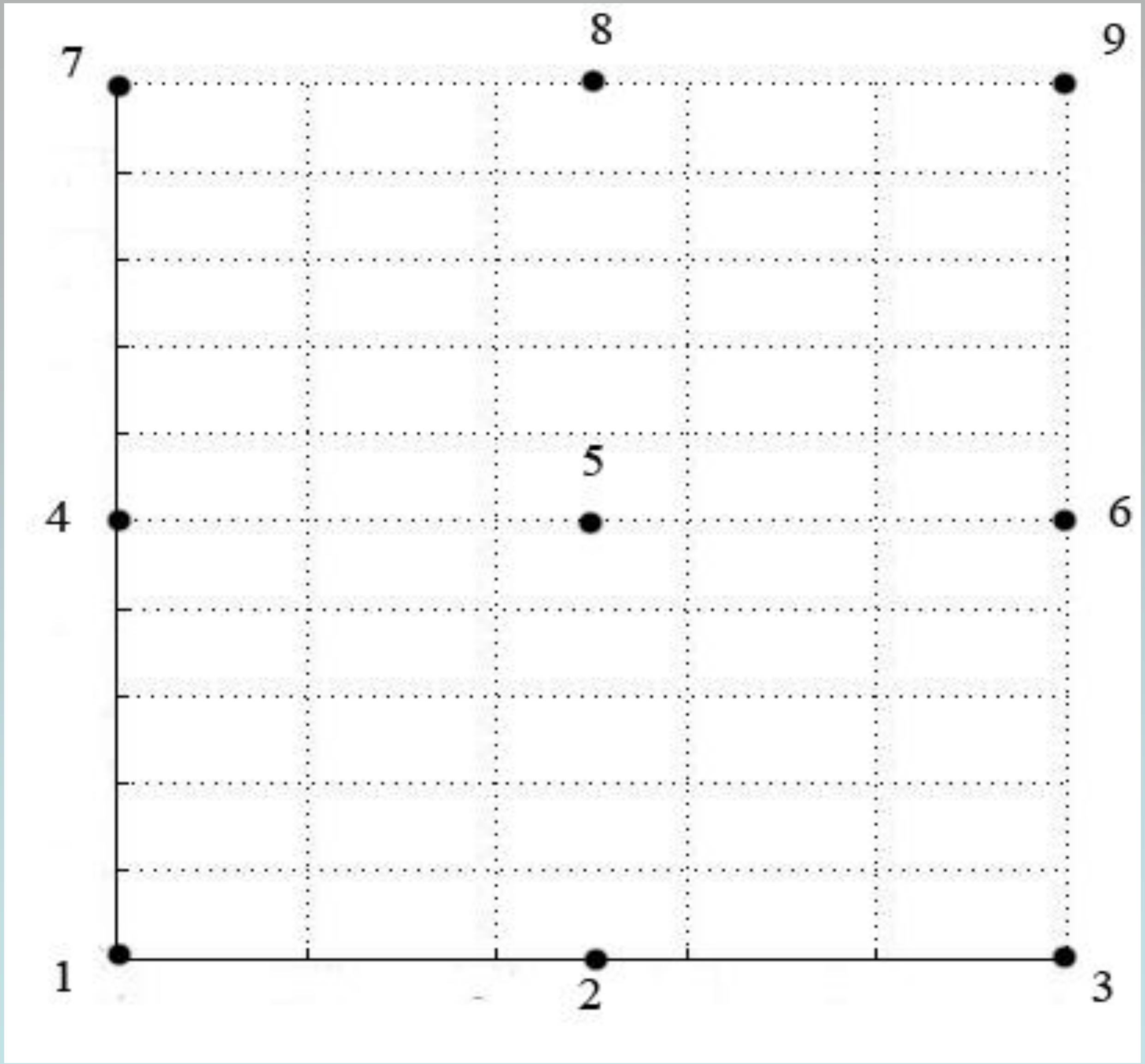
And the address of point C must be

213213.......

The game cycles through these points in this order.

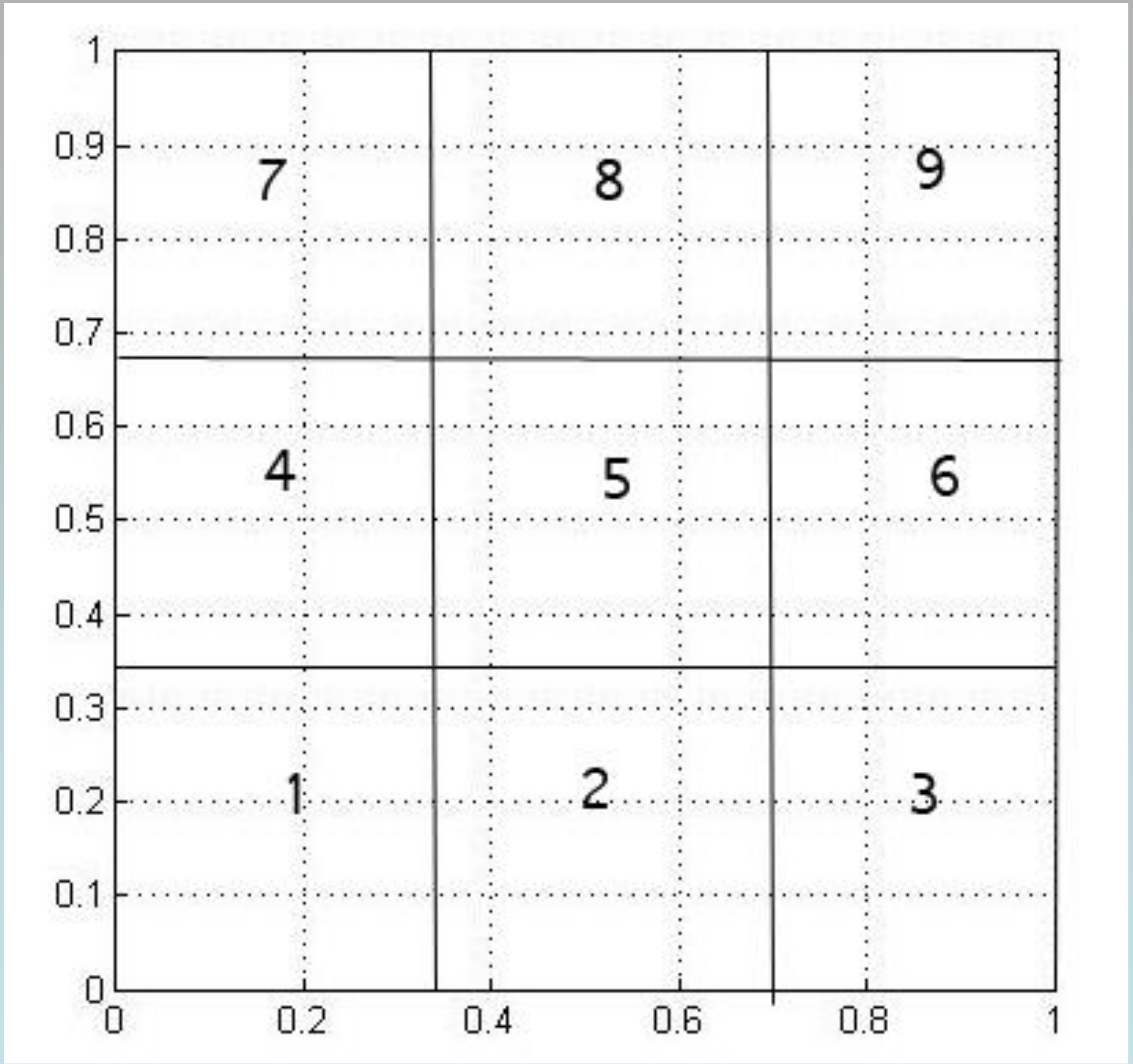
Testing sequences for randomness

For sequences of 1, . . . , 9

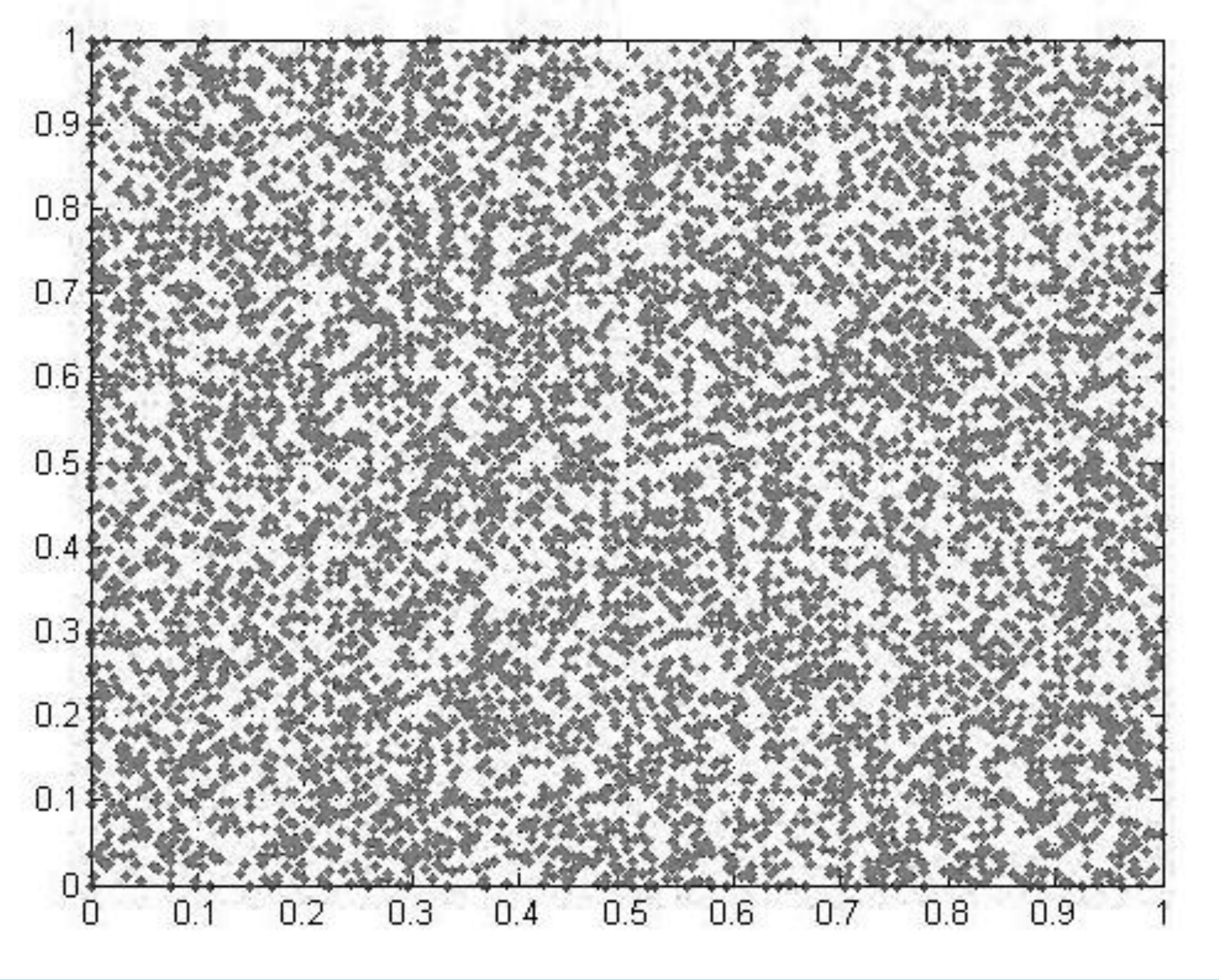


“Move 8/9 distance towards pin #k”

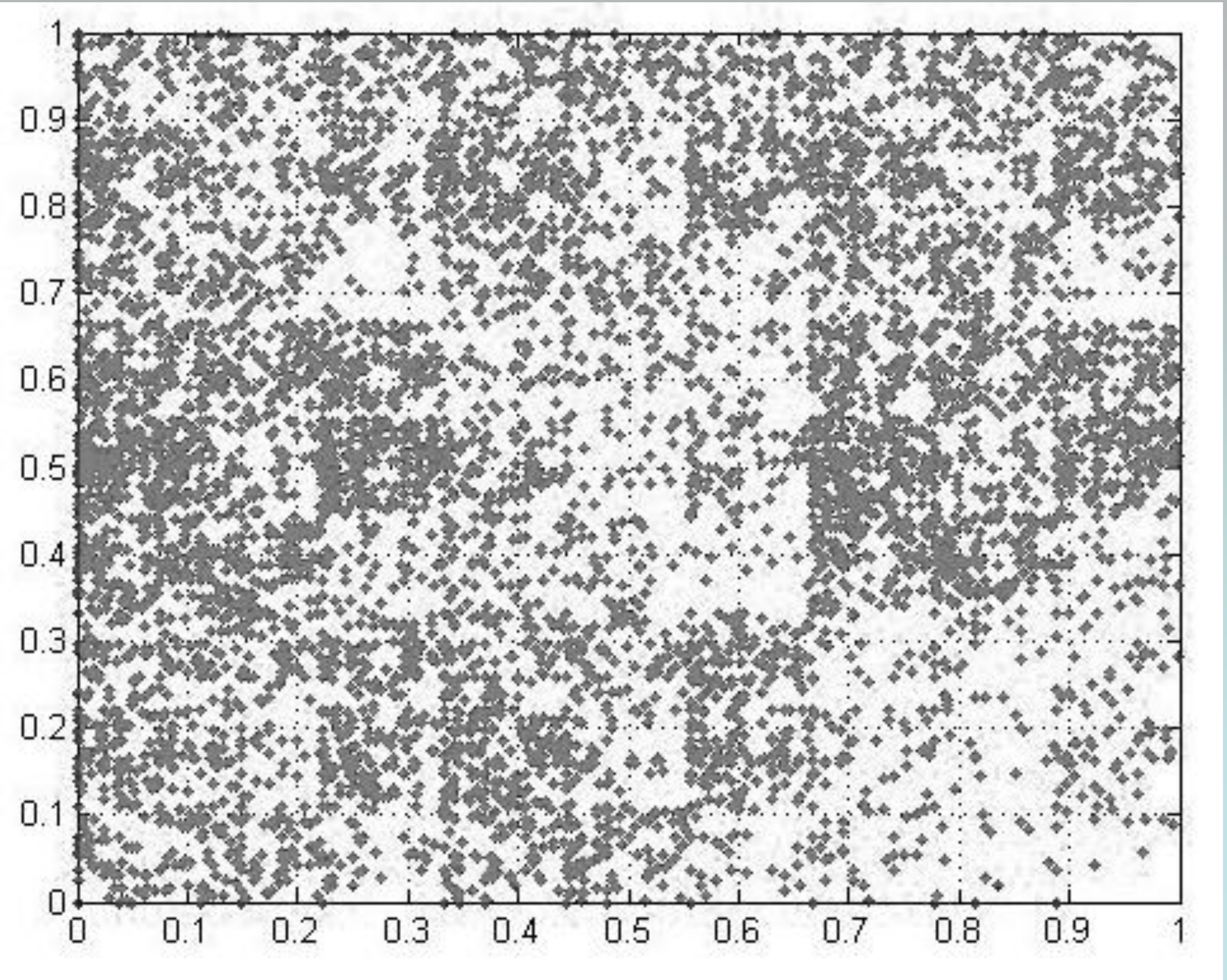
Addresses



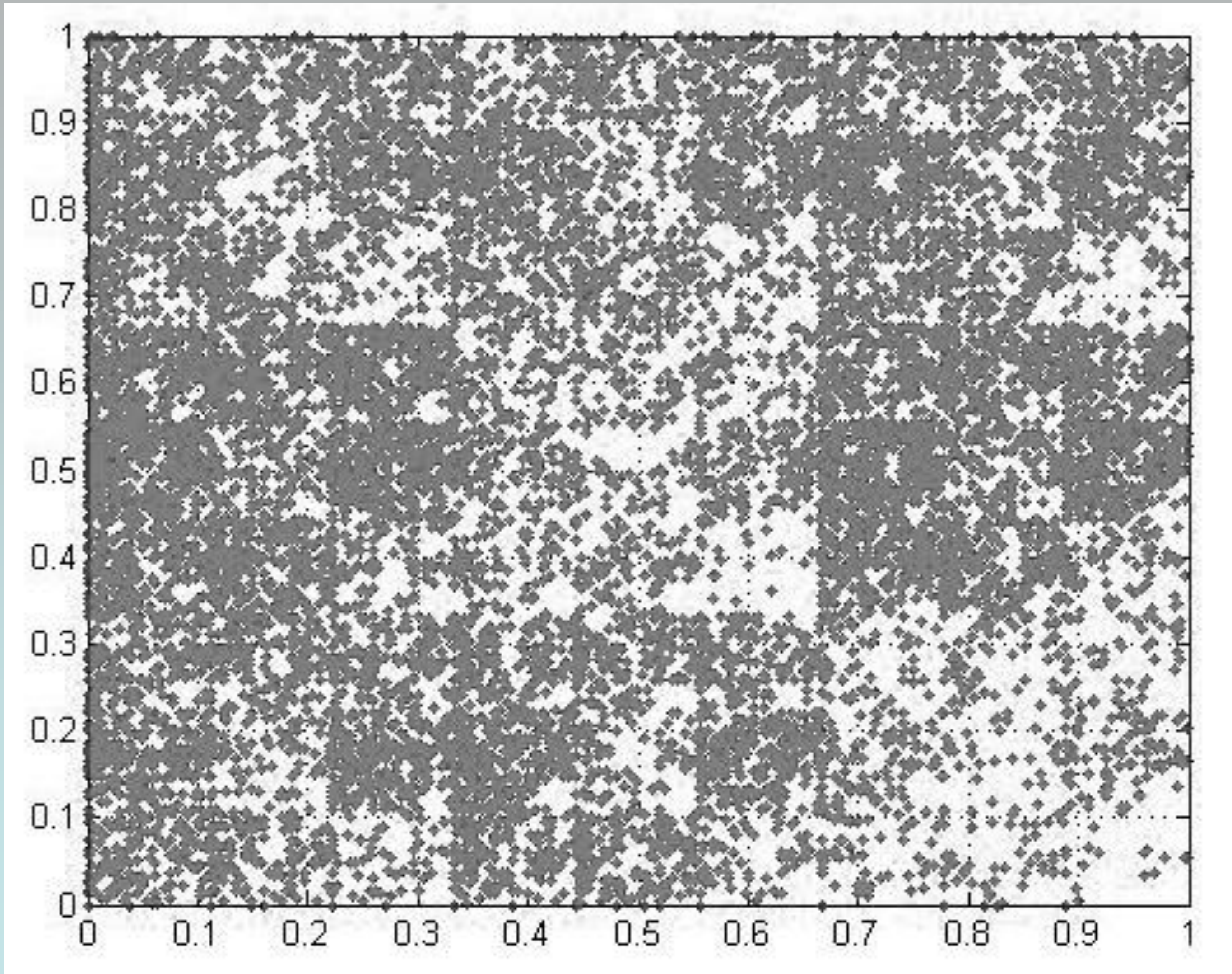
All digits occurring equally likely



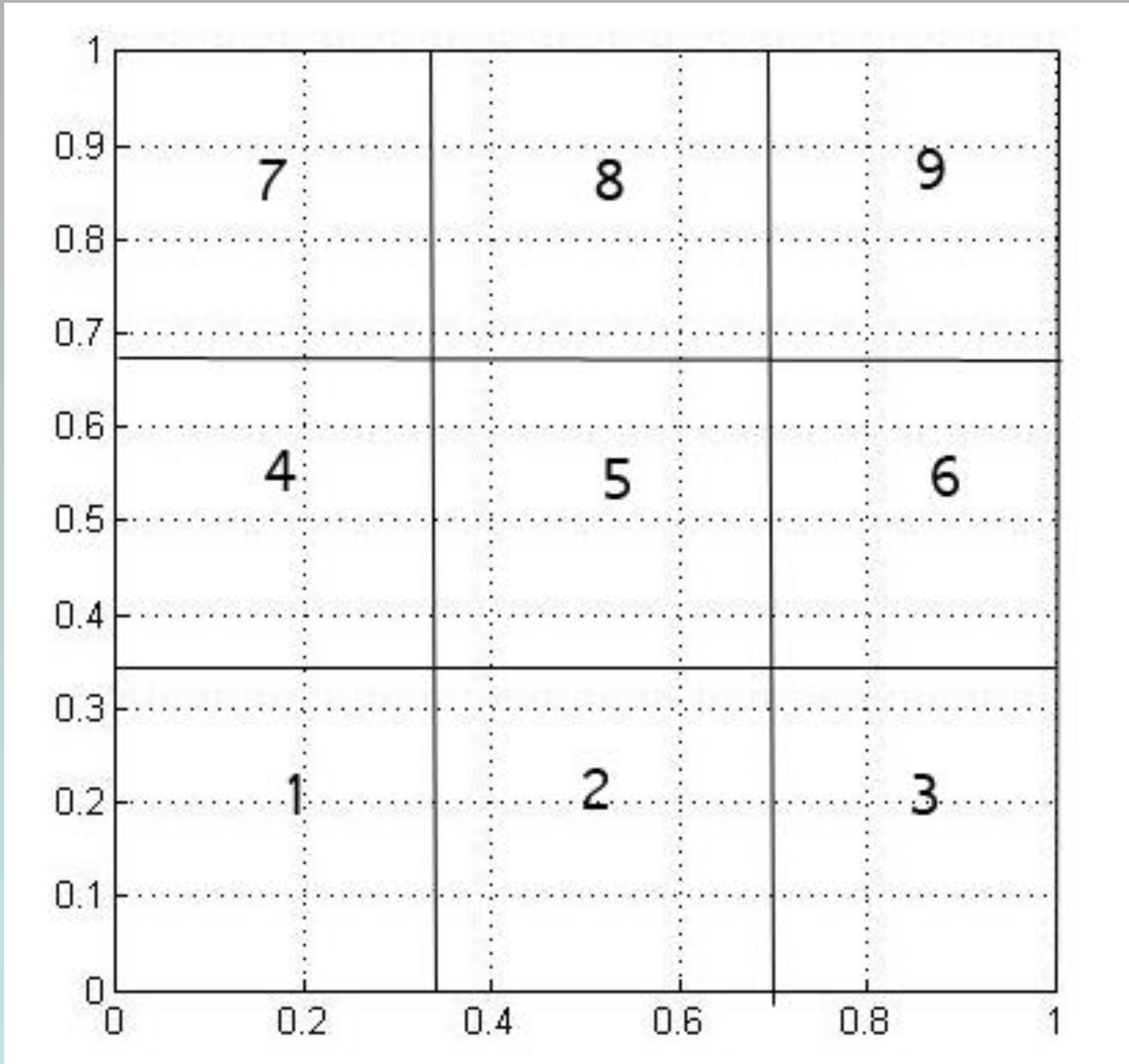
What are the probabilities for this sequence?



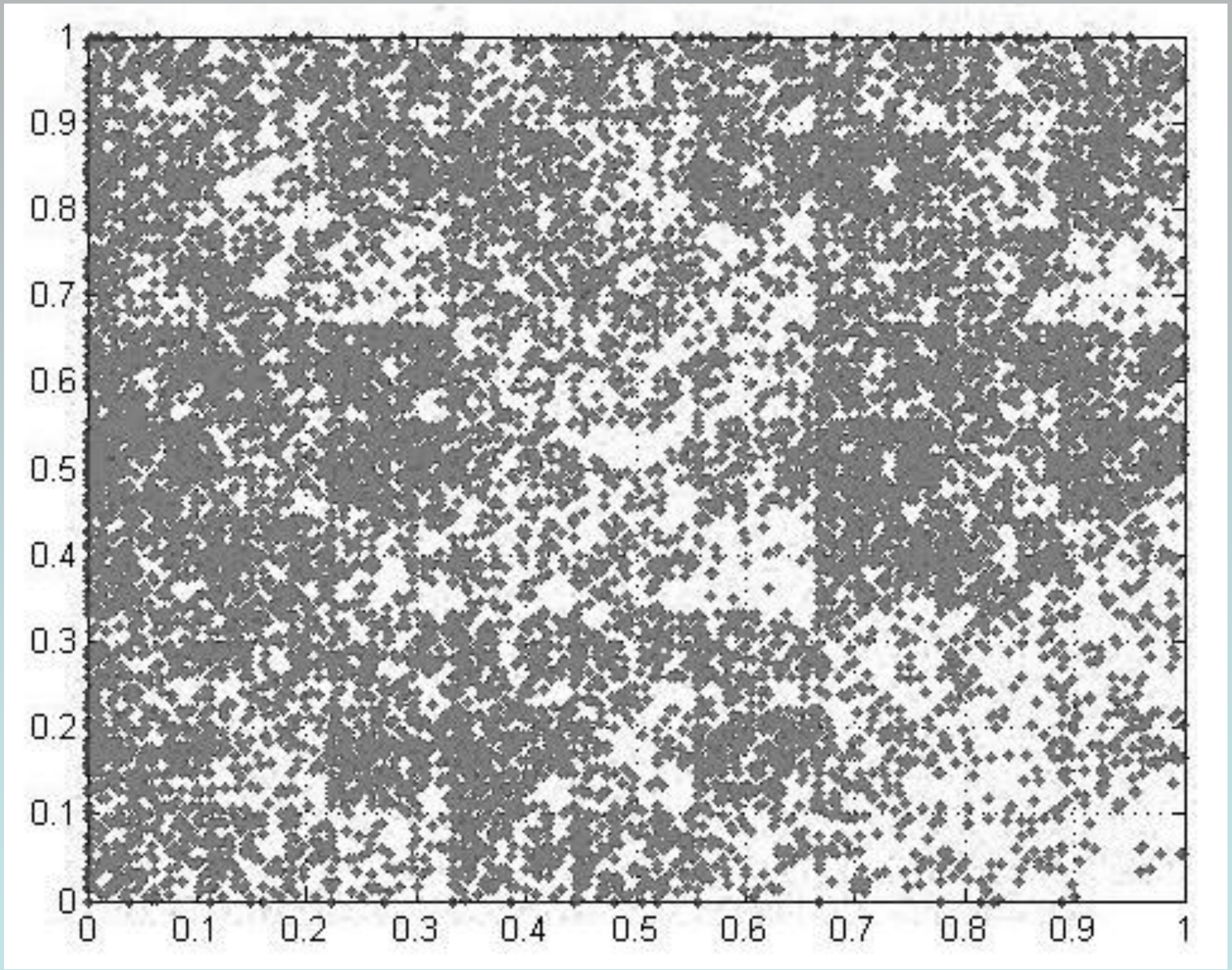
More points…..



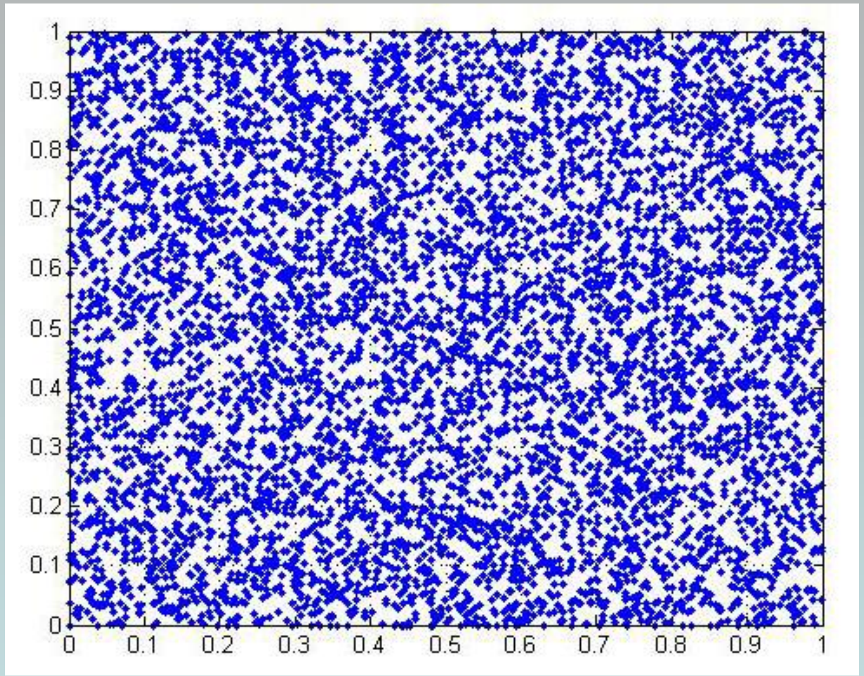
Addresses

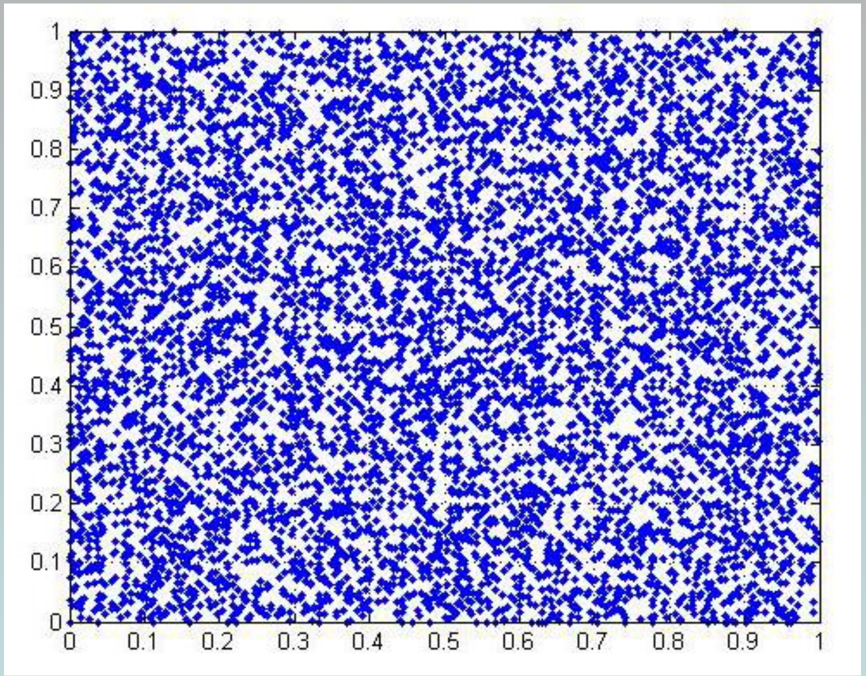


Looks like p3 is too low….



Testing sequences for randomness





10,000 random

10,000 digits of Pi

(digits 1, ..., 9)

Using the chaos game in bioinformatics

J.S.Almeida et al. "Analysis of genomic sequences by Chaos Game Representation", Bioinformatics, 2001

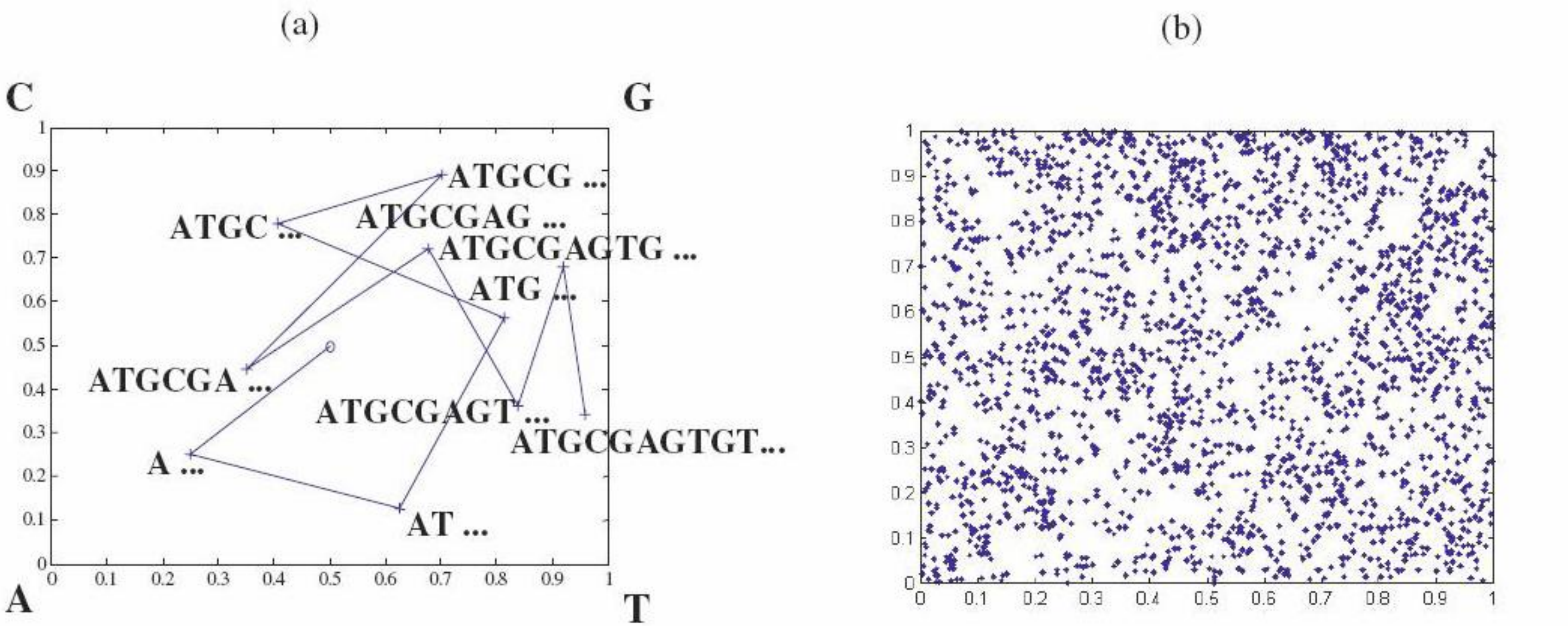


Fig. 1. (a) Chaos Game Representation (CGR) of the first 10 nucleotides of *E.coli* gene *thrA*: ATGCGAGTGT. The coordinates for each nucleotide are calculated iteratively using (0.5, 0.5) as an arbitrary starting position (equation 1). The pointer is moved half the distance to the next nucleotide to determine the next position (equation 1). (b) CGR of the full *thrA* sequence, totaling 2463 pairs of bases.

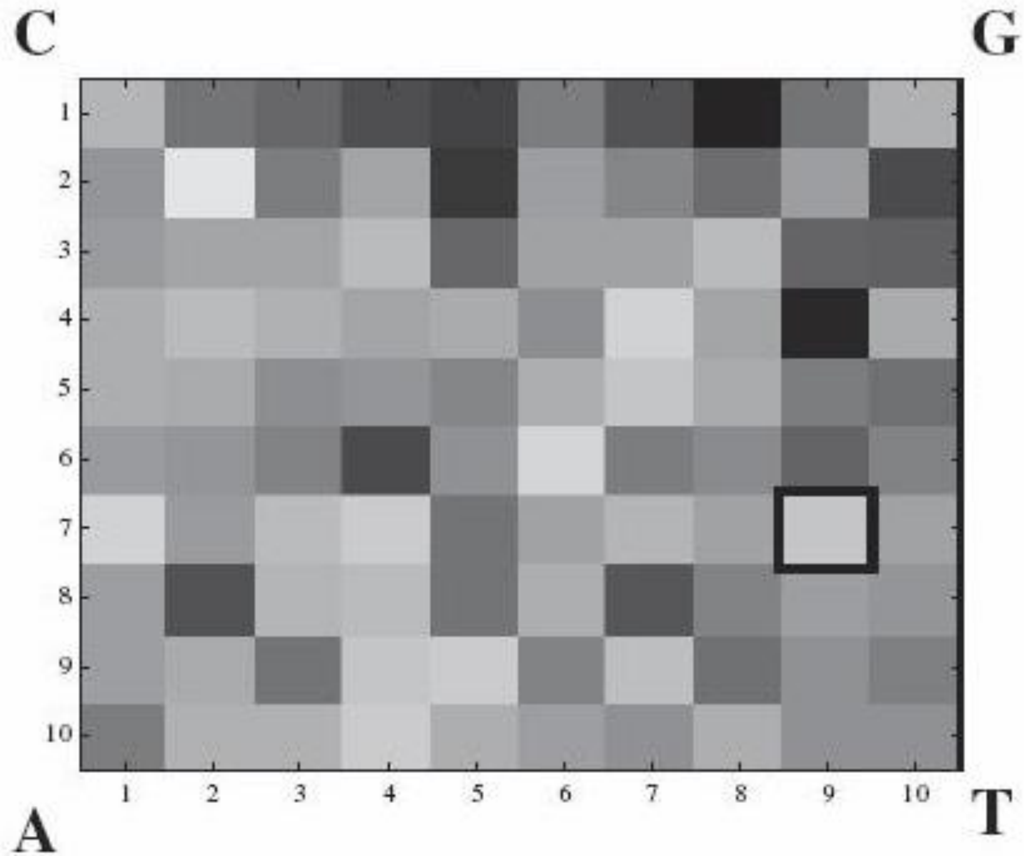
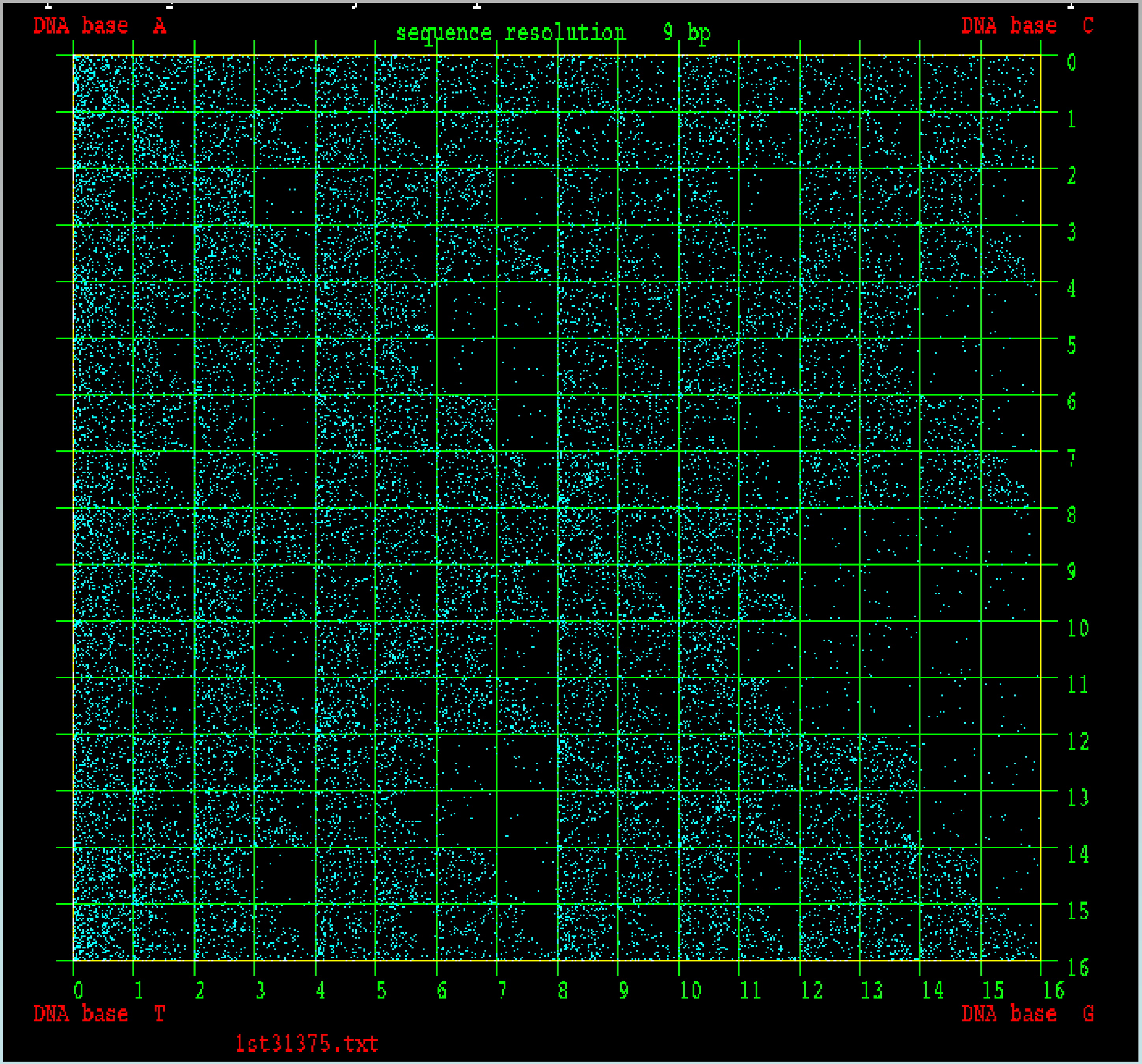
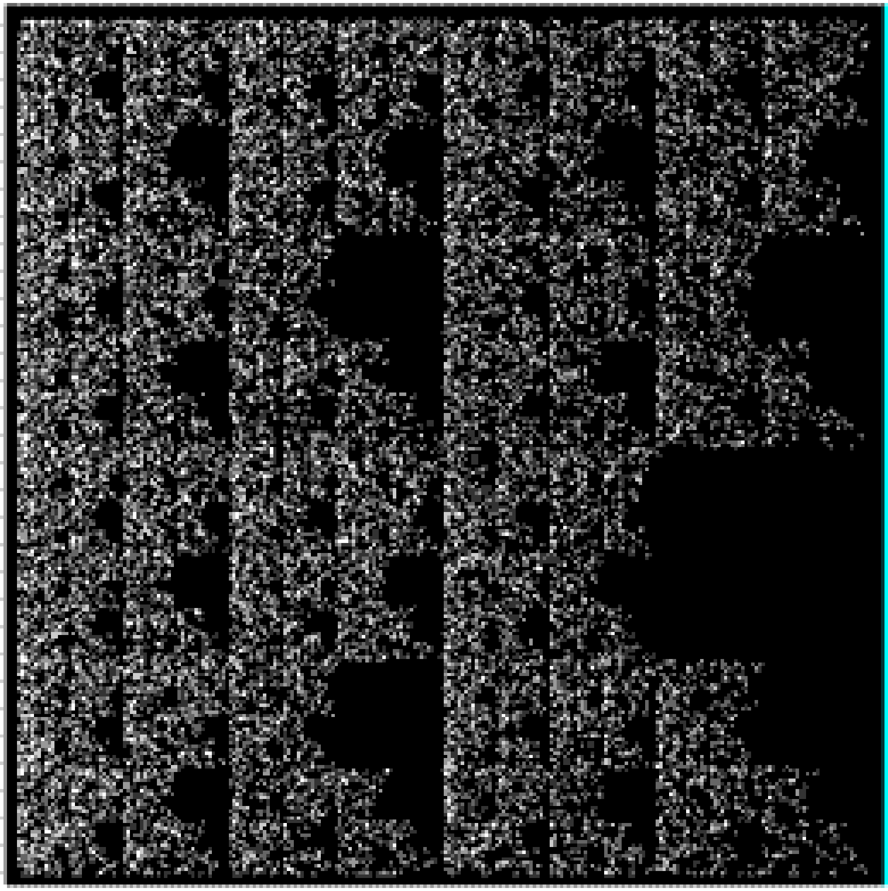


Fig. 5. $FCGR_{ThrA,3.32}$: frequency 10×10 table for CGR of *thrA*, $k = 100 \Rightarrow n = 3.32$. The gray scale represents frequencies between 0 (white) and the maximum frequency in any quadrant (black). The 8th position of *E.coli*'s *thrA* will now fall in the framed quadrant, delimited by `...(agga)aggaagt`; `...(cgta)cgtacgt`; `...(ggaa)ggaaggt`; `...(tgca)tgcatgt`.


DNA base A
sequence resolution 9 bp
DNA base C
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
DNA base T
DNA base G
1st31375.txt

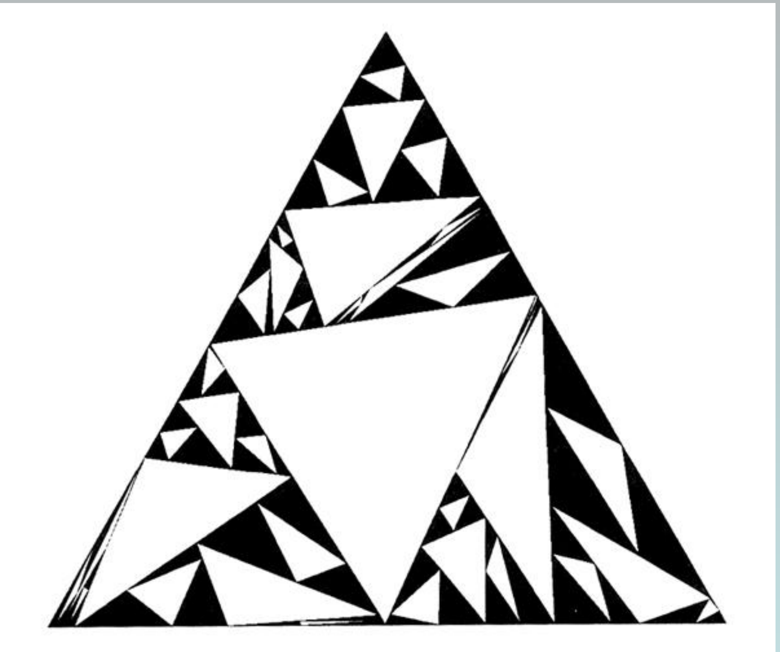


Stop | Full Square | Fastest | Addr. length: 1 | Sample

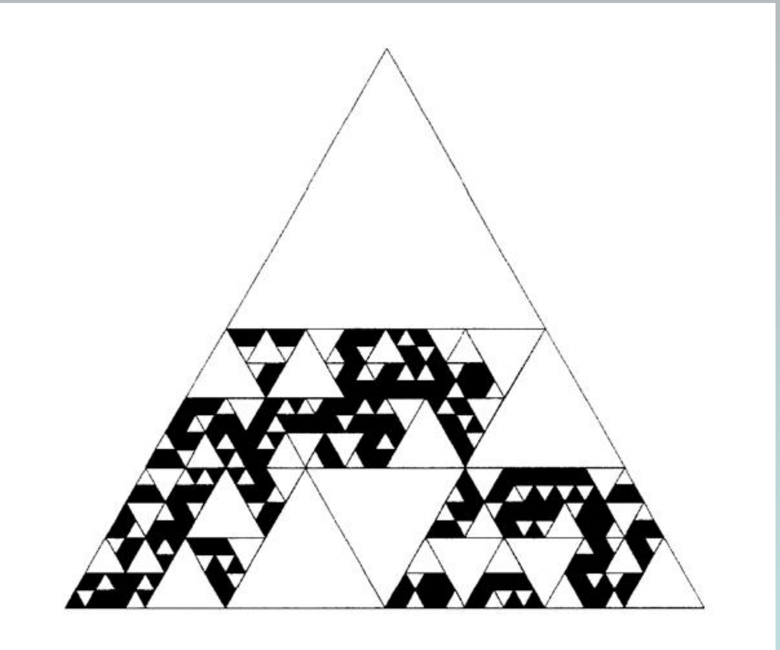
i	a	b	c	d	e	f	p
1	0.5	0.0	0.0	0.5	0.0	0.0	0.2998
2	0.5	0.0	0.0	0.5	0.0	0.5	0.3004
3	0.5	0.0	0.0	0.5	0.5	0.5	0.1968
4	0.5	0.0	0.0	0.5	0.5	0.0	0.2030
5							
6							
7							
8							
9							
10							

Dim: | # | Substring: 34

Random Fractals
(randomize the actions)

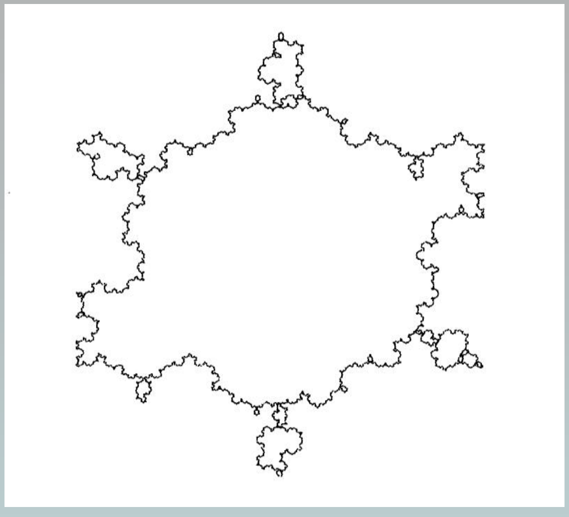


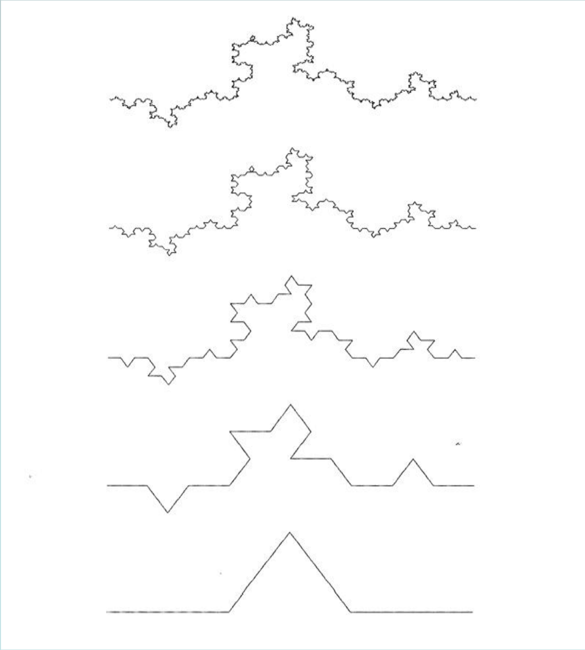
Random midpoints to
define triangle decomposition

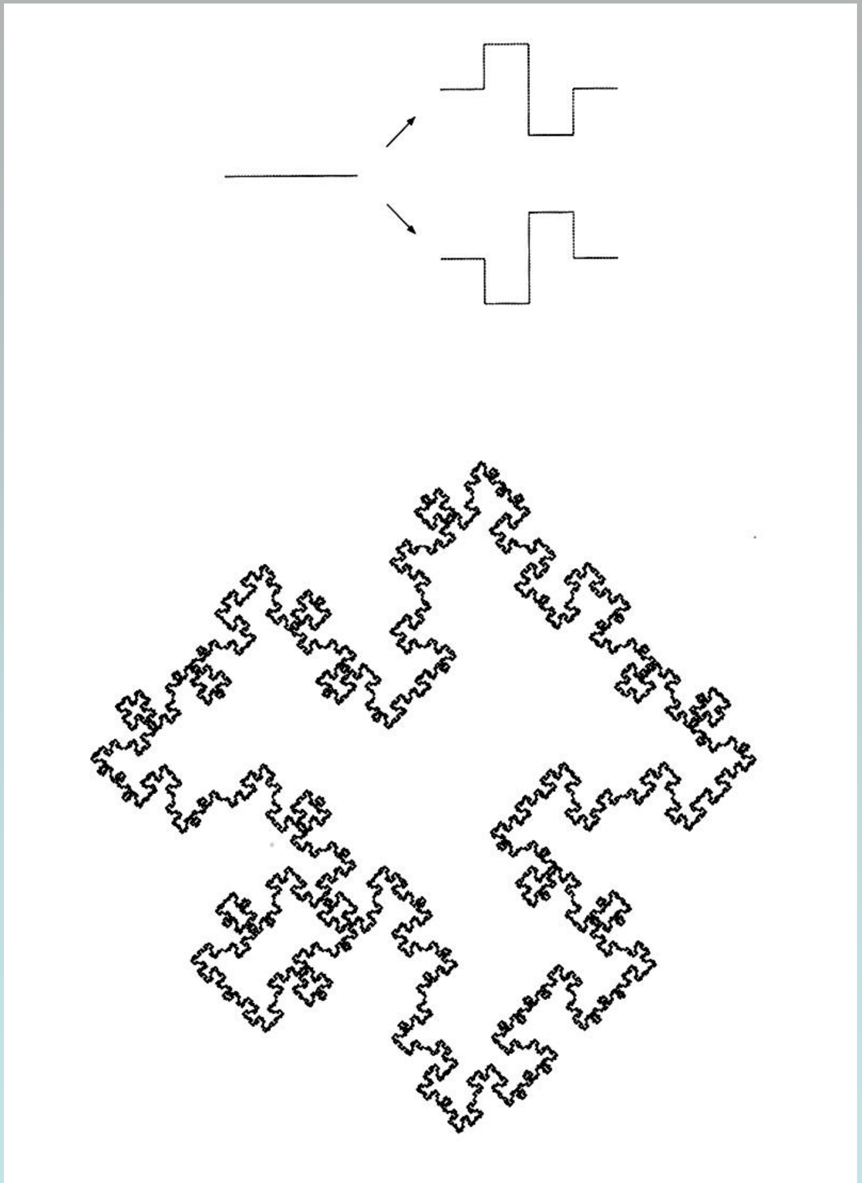


Remove random triangle
at each iteration

Random fractal curves





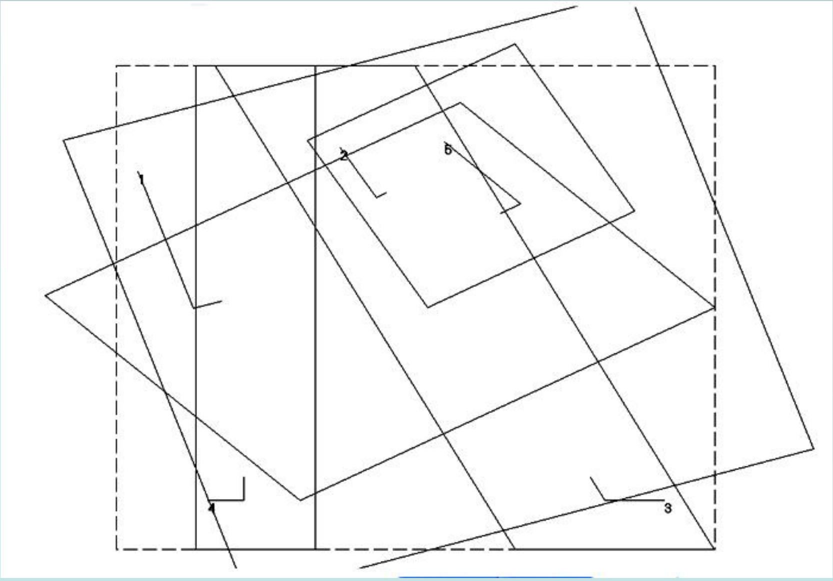


Using fractals to create real life images

Fractal Clouds

Fractal Clouds

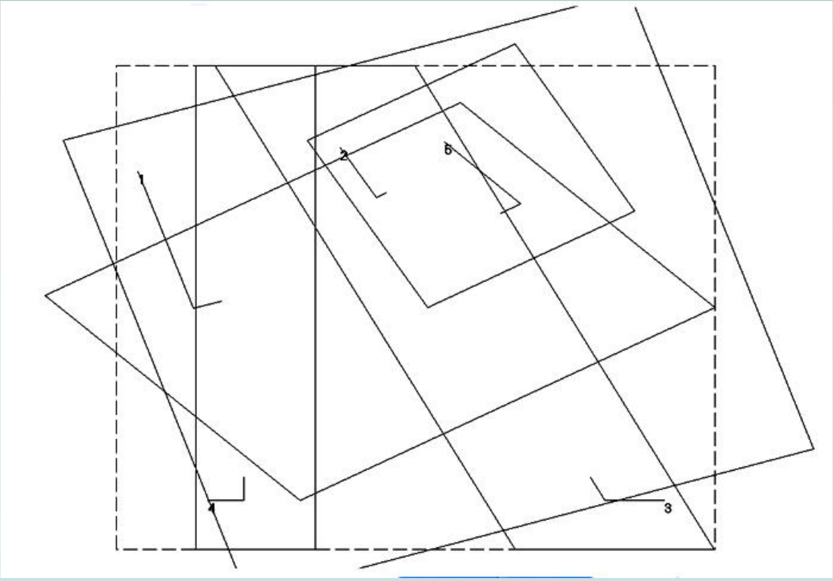
Decide the self-similar pieces

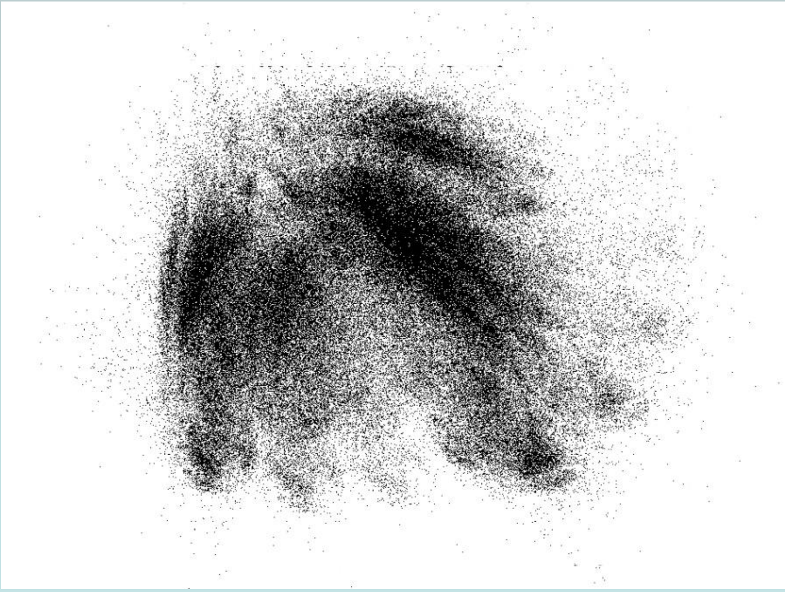


Fractal Clouds

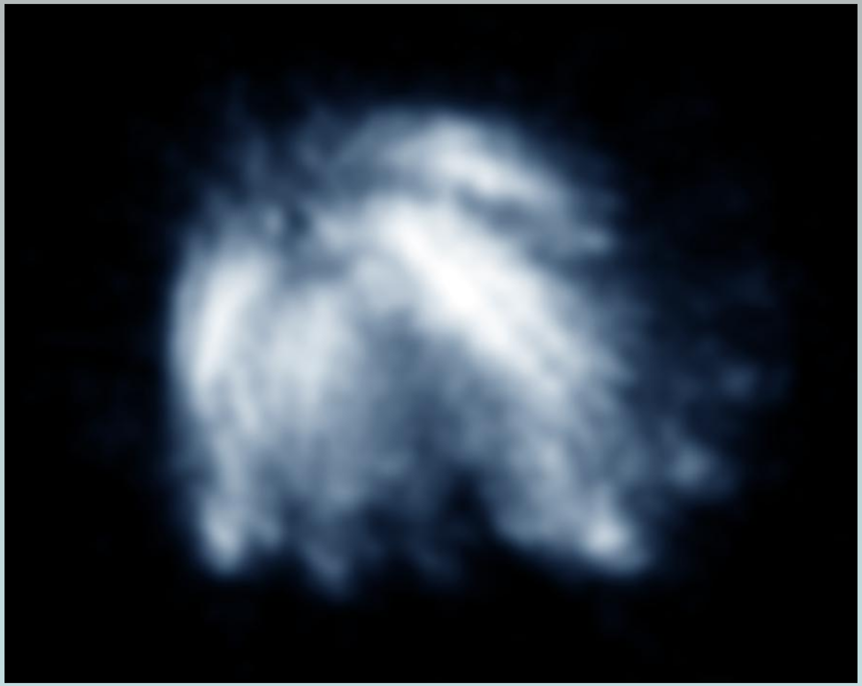
Decide the self-similar pieces

Generate the fractal



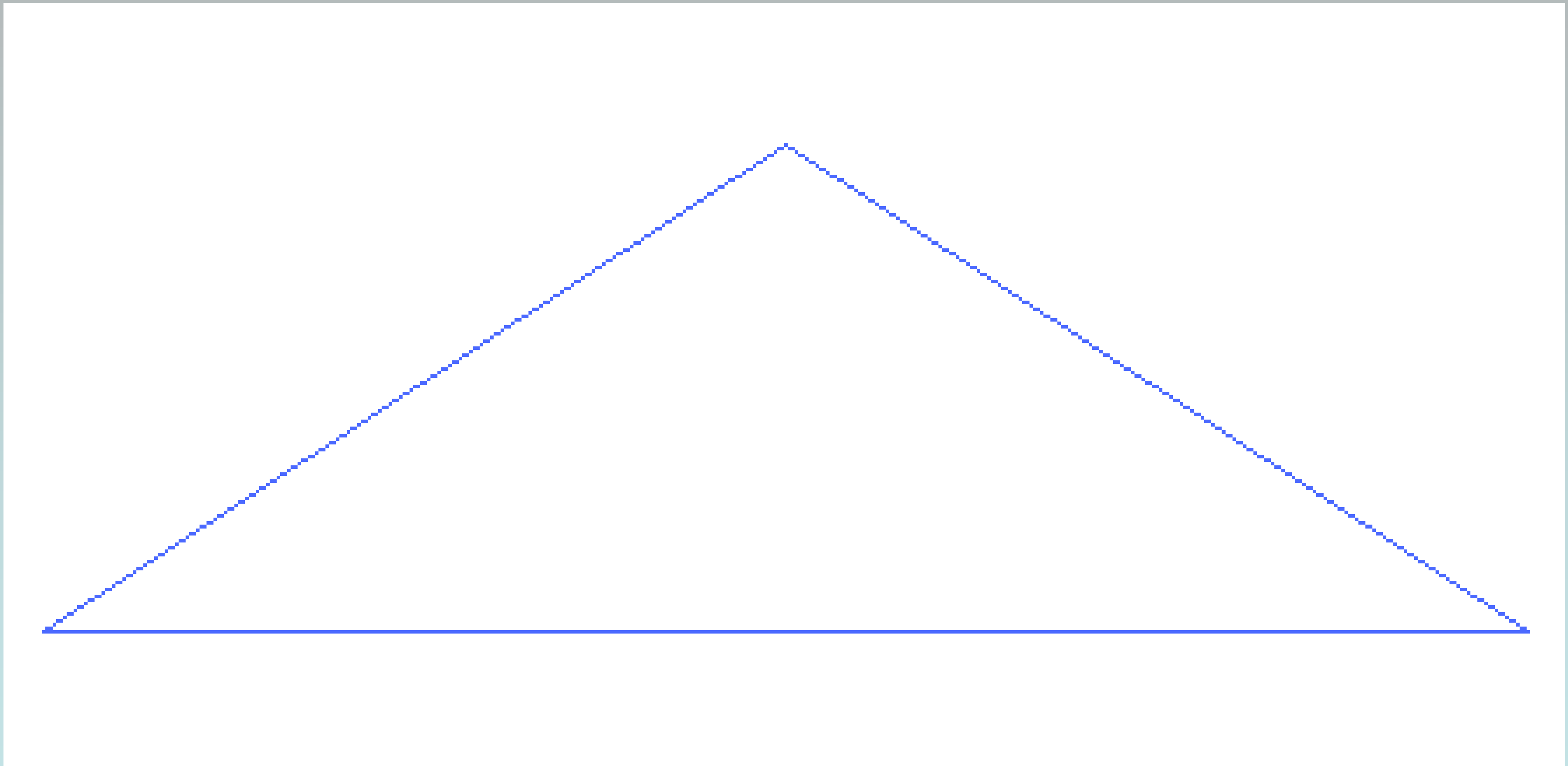


now blur a bit



Create a realistic cloud!

Creating fractal mountains…..



Fractal Mountains

Fractal image compression

Fractal image compression

Recall the IFS…….

$$W = w_1 \cup w_2 \cup \cdots \cup w_k, \qquad w_i = A_i + v_i$$

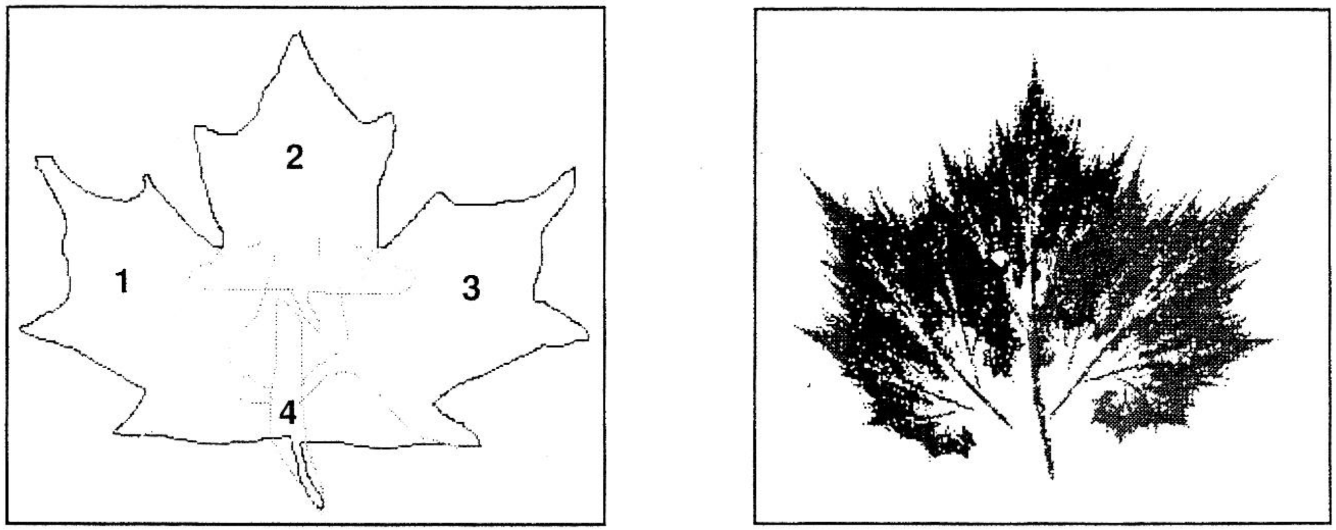
$$\lim_{n \to \infty} W^n(D) = \mathcal{F}$$

$$W(\mathcal{F}) = (\mathcal{F})$$

Fractal image compression

- A fractal can be reconstructed by playing the chaos game (or iteration of the IFS)
- One only needs the ‘game rules’ (or the affine transformations of the IFS)
- Equations need only ~ 10’s of KB
- Images need ~1,000 of KB!

Many fractals can be made to look like real-life images….



(a) (b)

Fig. 2.3.1 Collage Theorem example. (a) The original image and 4 subimages; (b) the attractor image.

Using a real image as a guide to finding an appropriate IFS

'Encoding' an IFS

… and many real life images look like (many) fractals

Figure 1.7: Self-similar portions of the Lenna image.

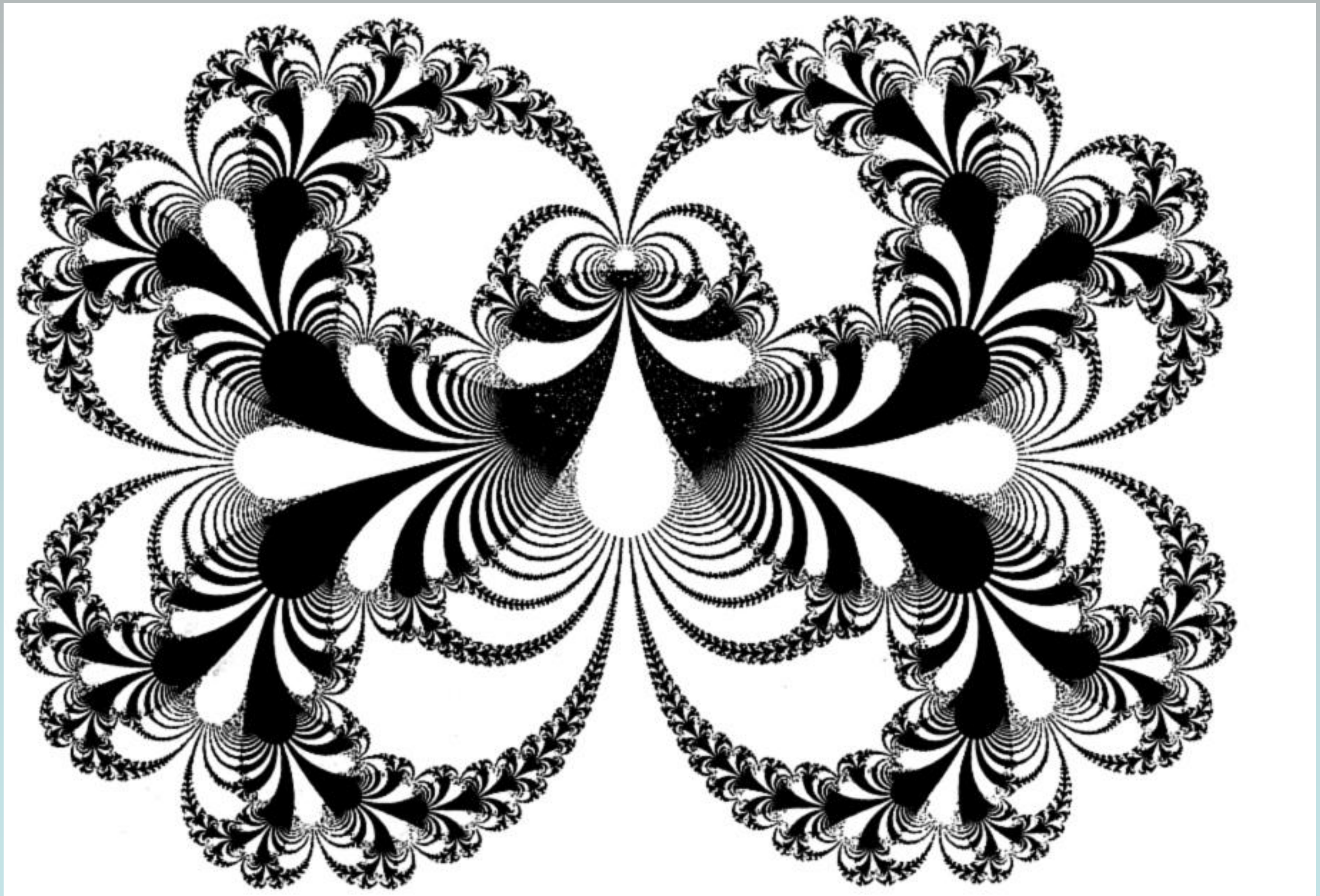
Determine which small pieces of the image can be used to re-create other pieces of the image.

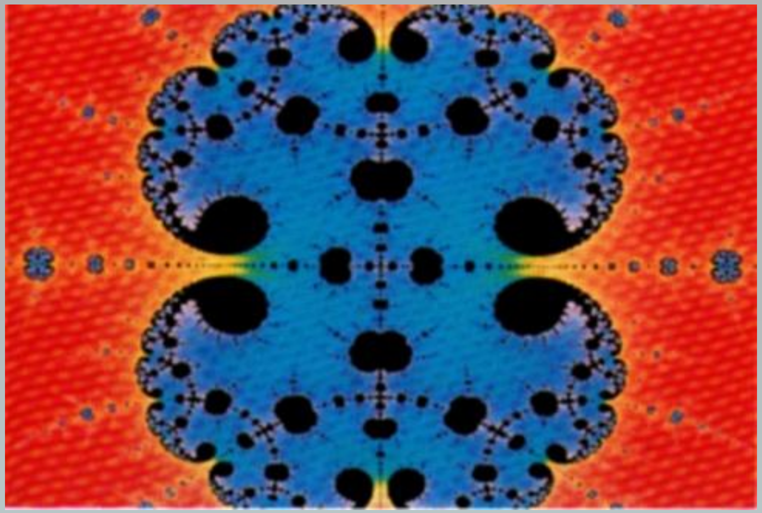
Collect all these IFS’s into a ‘multi IFS’ that will re-create the entire image.

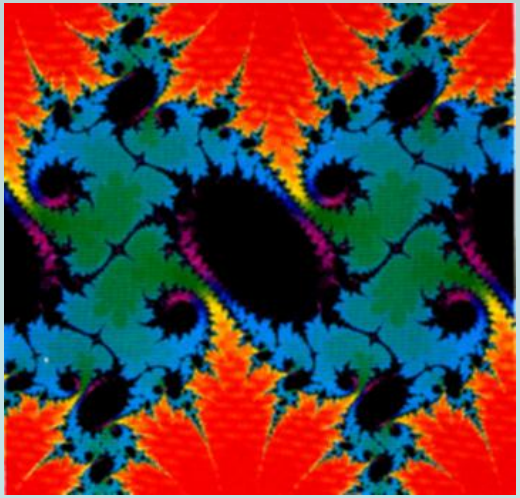
Figure 1.7: Self-similar portions of the Lenna image.

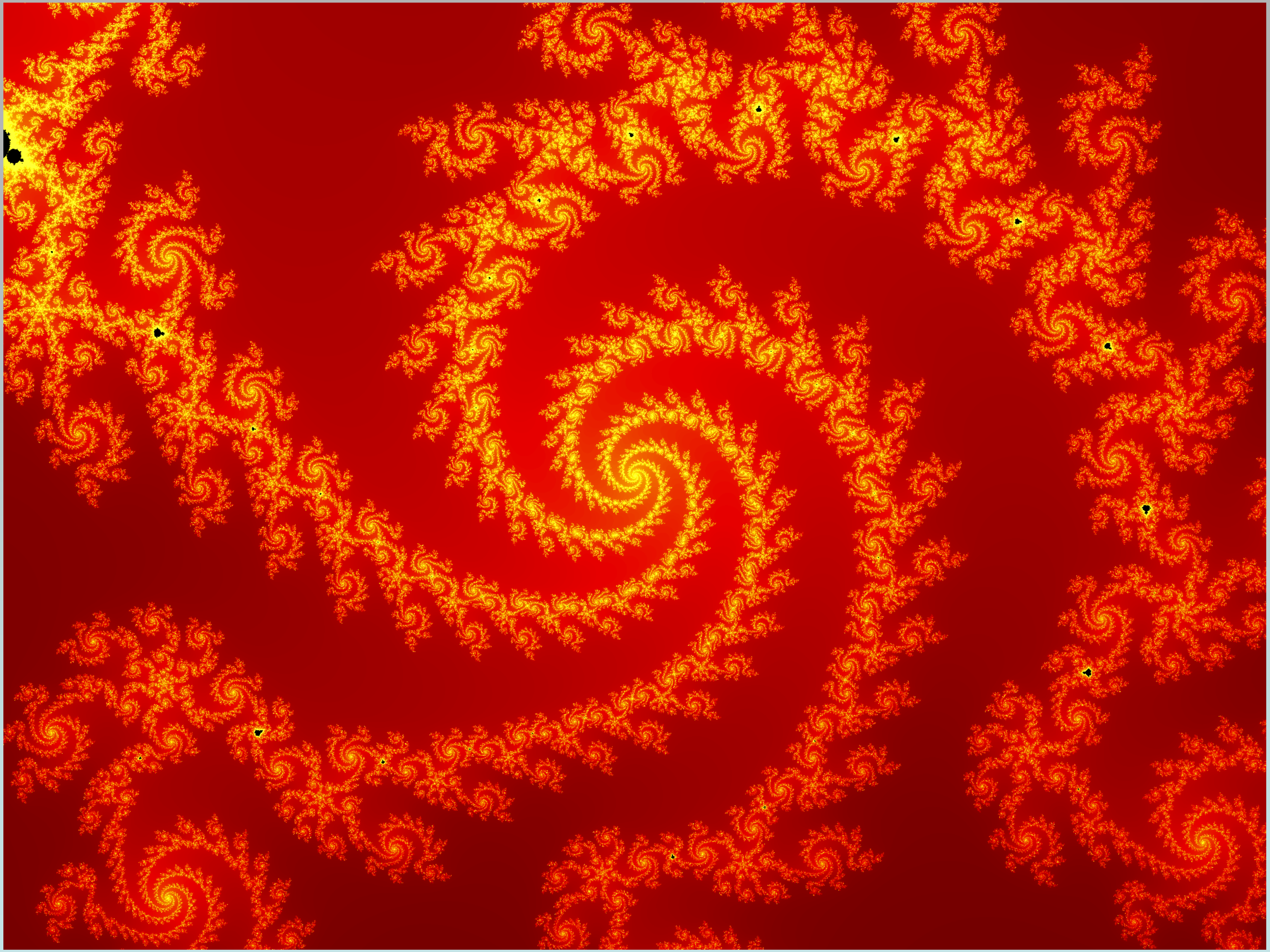
Julia sets

G. *Julia, P. Fatou ca 1920*

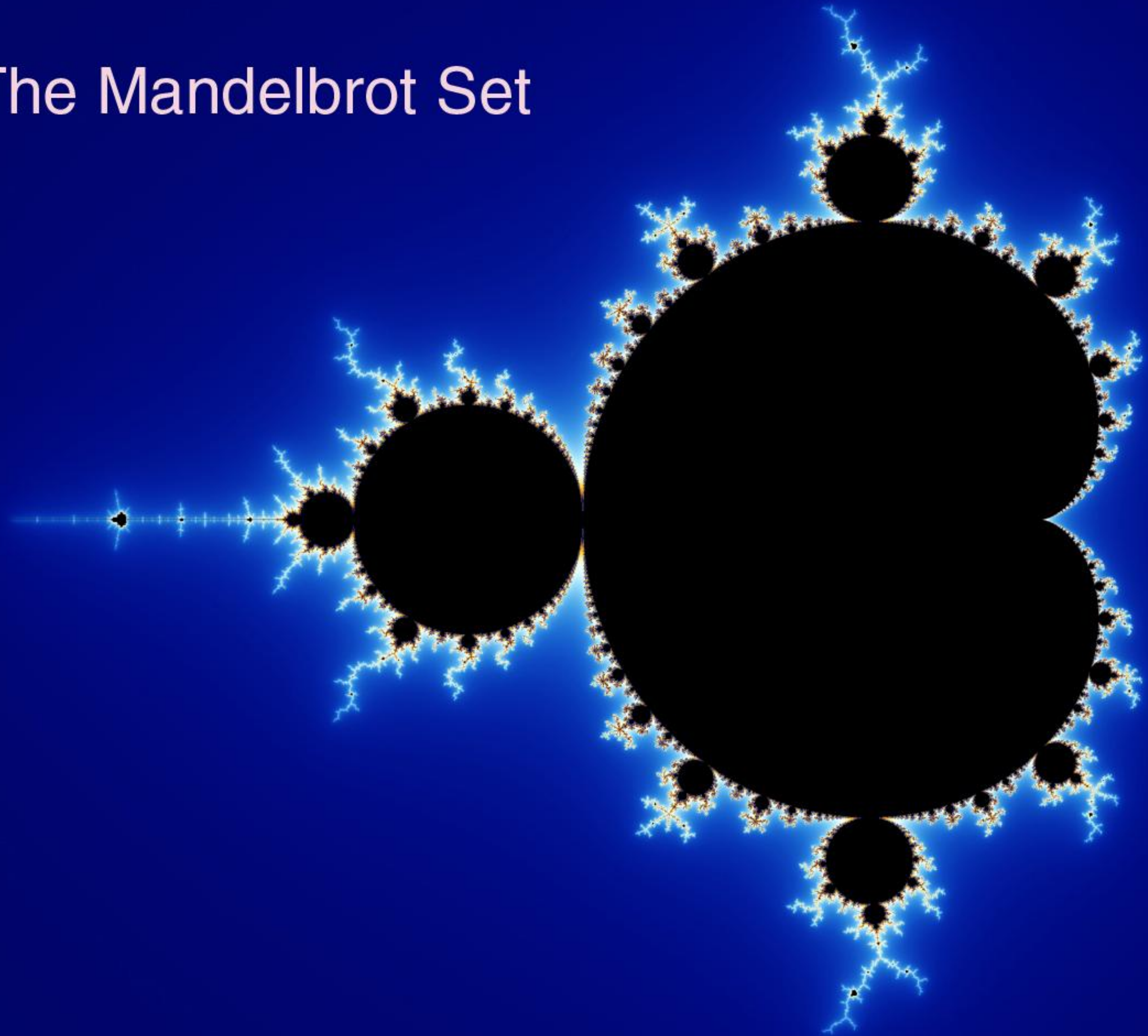


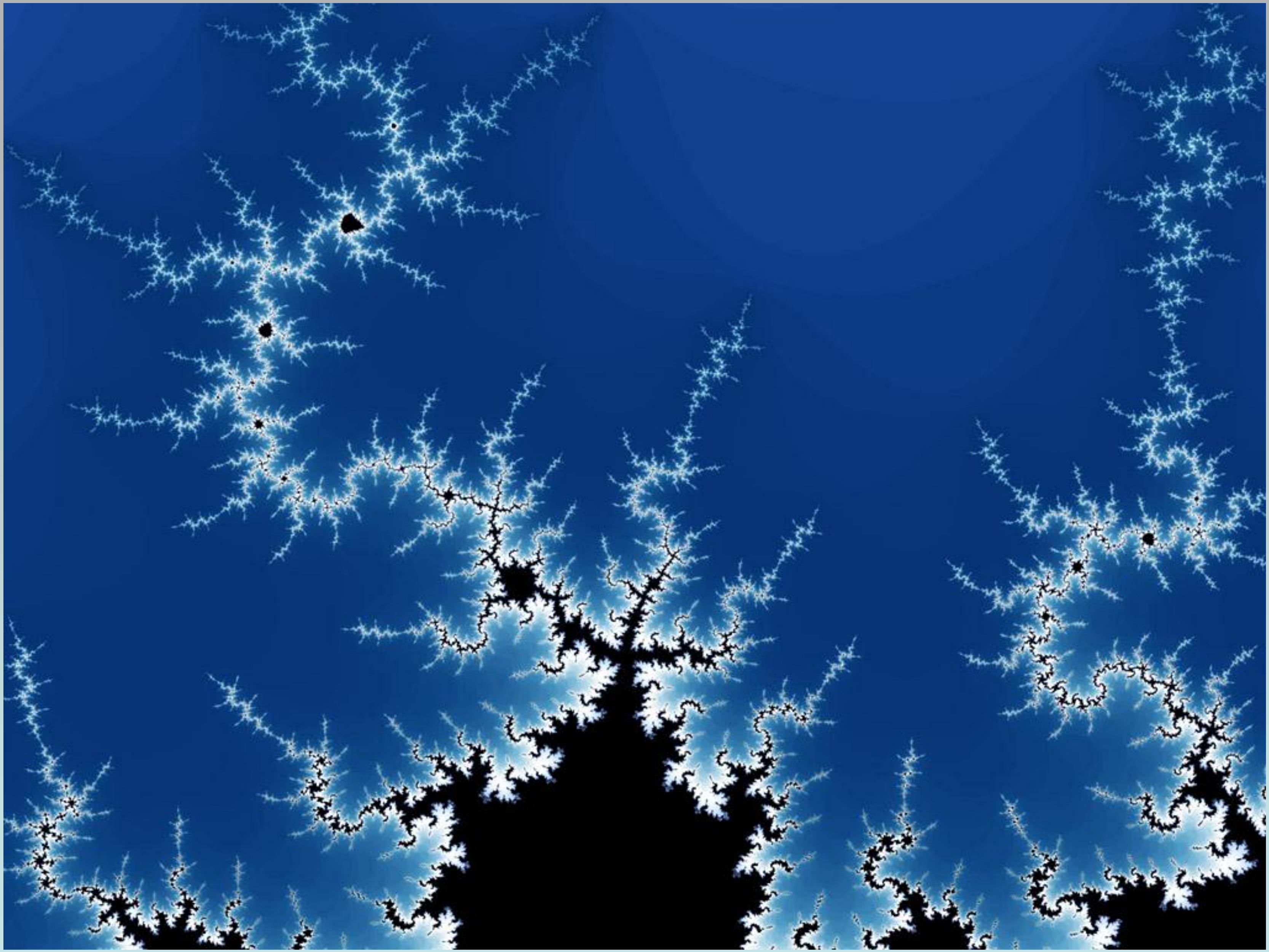




The Mandelbrot Set





For more information:

- *This presentation:* *www.sfu.ca/~rpyke/chaosgame.pdf*

- *More info:*

www.sfu.ca/~rpyke/ → “Fractals”

- *Email: rpyke@sfu.ca*